

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

People Re-Identification in Multi-camera Environments

Bruno Macedo Martins Santos Moreira

MSC DISSERTATION REPORT



Master in Electrical and Computers Engineering

FEUP Supervisor: Luís Corte-Real

Co-Supervisor: Pedro Carvalho

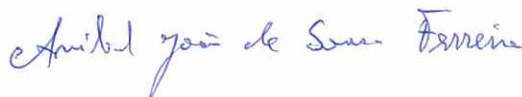
June 29, 2015

A Dissertação intitulada

“People Re-identification in Multi-camera Environments”

foi aprovada em provas realizadas em 20-07-2015

o júri



Presidente Professor Doutor Aníbal João de Sousa Ferreira
Professor Associado do Departamento de Engenharia Eletrotécnica e de
Computadores da Faculdade de Engenharia da Universidade do Porto



Professor Doutor José Manuel de Castro Torres
Professor Associado do Faculdade de Ciências e Tecnologia da Universidade
Fernando Pessoa



Professor Doutor Luís António Pereira de Meneses Corte-Real
Professor Associado do Departamento de Engenharia Eletrotécnica e de
Computadores da Faculdade de Engenharia da Universidade do Porto

O autor declara que a presente dissertação (ou relatório de projeto) é da sua exclusiva autoria e foi escrita sem qualquer apoio externo não explicitamente autorizado. Os resultados, ideias, parágrafos, ou outros extratos tomados de ou inspirados em trabalhos de outros autores, e demais referências bibliográficas usadas, são corretamente citados.



Autor - Bruno Macedo Martins Santos Moreira

Resumo

Com o aumento das necessidades de segurança e número de câmaras, surgiu também a necessidade de criar algoritmos capazes de lidar com o respectivo aumento do volume de informação. Num cenário de segurança, esta informação pode ser usada para detetar acessos a zonas restritas ou seguir os movimentos de uma pessoa numa área. Para tal, é necessário primeiro extrair informação relevante, que inclui determinar a localização das pessoas; de seguida, é necessário seguir a posição da pessoa enquanto esta se movimenta na cena; por último, quando ocorrem oclusões e movimentos entre câmaras, re-identificar as pessoas de forma a manter a sua identificação. Este processo é particularmente importante num cenário onde se pretende seguir uma pessoa numa cena, pois falhar a etapa de re-identificação significaria que o sistema de segurança deixaria de funcionar conforme previsto.

Esta dissertação tem como principal objetivo a criação de um algoritmo de re-identificação de pessoas que seja capaz de extrair informação relevante de forma a criar um modelo para a pessoa. A extração das características mais importantes não é uma tarefa trivial, pois surgem mudanças drásticas na forma como a pessoa é visualizada quando ocorrem oclusões ou na sua transição entre câmaras. A transição entre câmaras é particularmente desafiante pois nesses casos o algoritmo de seguimento não pode ser usado em auxílio da re-identificação, as pessoas aparecem muitas vezes em poses diferentes das anteriores e as câmaras podem estar sujeitas a condições de iluminação diferentes.

O modelo proposto combina a melhor seleção das características testadas e cria um modelo 3D da pessoa de forma a conseguir lidar com variações na sua pose. Um aspeto importante no modelo criado é que não é necessário aprender todas as poses possíveis, conseguindo criar algumas das partes desconhecidas do modelo. É ainda capaz de ter comportamentos distintos de acordo com a resolução das regiões de interesse onde se encontram as pessoas. O resultado é um algoritmo com melhores resultados que algoritmos do estado da arte em condições equivalentes.

Abstract

With the increasing demand for security and the number of cameras, there is also the need of creating algorithms that can handle the volume of information. In a security scenario, this information can be used to detect unwanted access to a certain area or track the movements of an individual in a certain area. In order to do this, several different steps are needed: (1) extract the relevant information, such as finding the location of the persons; (2) follow the movement of a previously detected person as he/she moves around the scene; (3) re-identify the persons to maintain their identity even when dealing with situations of occlusions and multiple cameras. This is especially important in a scenario where the objective is to follow a certain person through the scene. Failing the re-identification stage means that a person that was being followed is lost and the security system stops working as it's supposed to.

This thesis focuses on the objective of creating a reliable way of extracting the persons' characteristics and building a model for them. Extracting the most relevant characteristics is not always easy as the information in an image can change drastically when the person changes its' angle towards the camera, is partially occluded or moves to another camera. The transition from one camera to another is particularly troublesome, as it often means that the tracking algorithm loses track of the person and is incapable of providing any aid to the appearance model in re-identification. Also, persons may often appear in different poses when entering other cameras and the cameras themselves may have illumination changes.

The proposed appearance model combines the best selection of the tested features and creates a 3D model of the person so that it can deal with pose changes. One important aspect of the model is that it's not necessary to learn the person in every pose as it is capable of creating some of the missing parts of the model. It also makes use of the different resolutions of the persons to handle the situation more effectively. The result is an algorithm that outperforms state of the art algorithms when tested in the same conditions.

Agradecimentos

Deixo o meu profundo agradecimento a todos os que me acompanharam em mais uma etapa da minha vida: os meus pais, o meu irmão, que muito apoio me deram nestes anos.

A Alexandra Familiar, por tudo o que representa e pelo quanto me ajudou a melhorar o trabalho.

Ao meu Orientador Professor Doutor Luís Corte-Real e ao meu Supervisor Externo Doutor Pedro Carvalho pelas inúmeras correções e sugestões que muito contribuíram para esta dissertação. Ao Américo Pereira e Gil Coelho, do INESC, que sempre estiveram dispostos a discutir os vários problemas que foram surgindo.

Um último agradecimento a todos os meus amigos e família.

Bruno Moreira

A dissertação foi desenvolvida também no contexto e em colaboração com os projetos: Media Arts and Technologies (MAT), NORTE-07-0124-FEDER-000061, financiado pelo Programa Operacional Regional do Norte (ON.2 - O Novo Norte), sob o Quadro de Referência Estratégica Nacional (QREN), através do Fundo Europeu de Desenvolvimento Regional (FEDER), e por fundos nacionais através da agência de financiamento Portuguesa, Fundação para a Ciência e a Tecnologia (FCT); Project QREN 23277 RETAIL PRO, um projeto de I&D em co-promoção financiado pelo Fundo Europeu de Desenvolvimento Regional (FEDER) através do ON2 como parte do Quadro de Referência Estratégica Nacional (QREN), e gerido pela Agência de Inovação (ADI); QREN 33910 ARENA, um projeto de I&D financiado pelo Fundo Europeu de Desenvolvimento Regional (FEDER) através do Programa Operacional Regional do Norte (ON.2 - O Novo Norte) como parte do Quadro de Referência Estratégica Nacional (QREN), e gerido pelo IAPMEI - Agência para a Competitividade e Inovação, I.P.

“It’s a-me, Mario!”

Mario

Contents

Resumo	i
Abstract	iii
1 Introduction	1
1.1 Contextualization	1
1.2 Objectives	2
1.3 Contributions	2
1.4 Document Outline	2
2 State of the Art	3
2.1 Concepts and System View	3
2.2 Classification	5
2.3 Appearance Models	7
2.3.1 Types of Features	7
2.3.2 Model Approaches	11
2.3.3 Object Correspondence	11
2.4 People Tracking in Multi-Camera Environments	13
2.4.1 Foreground Segmentation	13
2.4.2 People Detection	16
2.4.3 Object Tracking	18
3 Datasets and Assessment	21
3.1 Tracking Datasets	21
3.2 Re-Identification Datasets	22
3.3 Metrics	24
4 Individual Features	27
4.1 Feature Overview	27
4.1.1 Global Information	27
4.1.2 Local Features	31
4.2 Testing Overview	33
4.2.1 Global Information	34
4.2.2 Local Information	42
4.2.3 Result Overview	43
4.3 Detailed Analysis	45
4.3.1 Global Information	46
4.3.2 Local Information	61
4.4 Conclusions	63

5	Proposed Appearance Model	67
5.1	Initial Model Overview	67
5.2	Best Weight Combination	68
5.3	Local Features	71
5.4	Resolution Driven Appearance Model	72
5.4.1	Model Structure	72
5.4.2	Model Comparison	73
5.4.3	Results	74
5.5	Multi-Dimension Model	75
5.5.1	Pose Identification	76
5.5.2	Results	76
5.6	3D Model Extension	79
5.6.1	Interpolation Algorithm	79
5.6.2	Angle Variations	79
5.7	Low Resolution Problems	85
5.8	Learning Model	87
5.8.1	Decision Tree	87
5.8.2	$Threshold_{NewModel}$	87
5.8.3	$Threshold_{HighConfidence}$	88
5.8.4	$Threshold_{UpdateModel}$	89
5.9	Appearance Model Results	90
6	Conclusions	93
6.1	Final Discussion	93
6.2	Future Work	94
	References	95

List of Figures

2.1	Single Camera Tracking System	4
2.2	Multiple Camera Tracking System	4
2.3	Support Vector Machines example for a two dimensional space	5
2.4	Example of Application of the Brightness Transfer Function	8
2.5	Extracted Shape Patterns	9
2.6	Puppet-like Representation of the Person	9
2.7	Example of the use of Gaussian Models to find and Match Entry and Exit Points from the scene	12
2.8	Background Subtraction Process with N frames for background initialization and B and I being the background and frame images	13
2.9	Application of the Frame Differencing method	14
2.10	Example of a result from the ViBe Method	15
2.11	Example of Application of the LBP Method on a Patch	16
2.12	People Detection Framework with Visibility Probabilities	17
2.13	AdaPT General Framework	19
3.1	Annotated frame example from the CAVIAR dataset	22
3.2	Frame example over two cameras from the CAVIAR dataset	22
3.3	Examples from the PETS 2001 dataset	23
3.4	Example from the VIPeR Dataset	23
3.5	Examples from the CAVIAR4REID dataset	24
4.1	Example of the 3 Body Part Division, from the CAVIAR4REID Dataset	28
4.2	Example of Application of CENTRIST to a 3x3 Window	29
4.3	Edge Detectors for Edge Detection on 90°, 0°, 135° and 45°	30
4.4	Re-Identification Example	33
4.5	Example Grayscale Images Using Ellipse Region of Interest	34
4.6	CMC Curve for Grayscale Features	35
4.7	CMC Curve for 9 RGB Histograms	37
4.8	CMC Curve for HSV Features	38
4.9	CMC Curve for CENTRIST Features	39
4.10	CMC Curve for Wavelet Features	40
4.11	CMC Curve for Edge Energy Feature	40
4.12	CMC Curve for Laplacian Features	41
4.13	Visual Confusion Matrix Representation for Grayscale Features	46
4.14	1 st Rank Result for Each Person Using 3 Body Part Grayscale Histogram	47
4.15	Best Person for Grayscale Re-Identification	48
4.16	Worst Person for Grayscale Re-Identification	48

4.17	Visual Confusion Matrix Representation for RGB Features	49
4.18	1 st Rank Result for Each Person Using 3 Body Part RGB Histogram	50
4.19	Best Person for RGB Re-Identification	51
4.20	Worst Person for RGB Re-Identification	51
4.21	Visual Confusion Matrix Representation for HSV Features	52
4.22	Example of persons confused with HSV features	53
4.23	1 st Rank Result for Each Person Using 3 Body Part HSV Histogram	54
4.24	Visual Confusion Matrix Representation for CENTRIST Features	55
4.25	1 st Rank Result for Each Person Using 3 Body Part CENTRIST Histogram	56
4.26	Best Person for CENTRIST Re-Identification	57
4.27	Worst Persons for CENTRIST Re-Identification	57
4.28	Visual Confusion Matrix Representation for Laplacian Features	58
4.29	1 st Rank Result for Each Person Using 3 Body Part Laplacian Histogram	59
4.30	Worst Person for Laplacian Re-Identification	60
4.31	Visual Confusion Matrix Representation for SIFT Features	61
4.32	1 st Rank Result for Each Person Using SIFT Features with the FAST Detector . .	62
4.33	1 st Rank Result for Each Person and Each Method	64
4.34	1 st Rank Result for Each Person and Each Method	65
5.1	Overview of the Appearance Model	68
5.2	CMC Curve for Color Features	69
5.3	CMC Curve for Texture Features	70
5.4	CMC Curve for Texture Features, Color Features and Mixed Model. The Mixed Model and Color Features almost overlay	70
5.5	Overview of the Appearance Model	71
5.6	CMC Curve For Texture, Color and Local Features	72
5.7	Appearance Model Extension, with High and Low Resolution	73
5.8	Model Creation Process	73
5.9	Resolution Driven Comparison Weights	74
5.10	CMC Curve For Texture and Color Features with Resolution Driven Comparison Weights	75
5.11	Different Poses for a Single Person in the CAVIAR4REID Dataset	75
5.12	CMC Curve with Pose Ground-Truth and Available Learned Pose (Single Pose) .	77
5.13	CMC Curve with Pose Ground-Truth, Available Learned Pose and Comparisons to Same Pose Only	77
5.14	CMC Curve with Pose Ground-Truth and Comparisons to Same Pose Only	78
5.15	CMC Curve with Pose Ground-Truth, All Pose Models and Comparisons to Same Pose Only	78
5.16	Generic 3D Appearance Model: Instead of using 3 views, the model can adapt to a generic n views	79
5.17	CMC Curve with Pose Ground-Truth, Interpolated Missing Poses, Comparisons to Pose and When Pose is Available	81
5.18	CMC Curve with Pose Ground-Truth, Interpolated Missing Poses, Comparisons to Pose and When Pose or Adjacent Pose is Available	81
5.19	CMC Curve with Pose Ground-Truth, Interpolated Missing Poses, Comparisons to All Poses	82
5.20	CMC Curve with Pose Ground-Truth, Interpolated Missing Poses, Comparisons to Same Pose and Adjacent	82

5.21 CMC Curve with Pose Ground-Truth, Interpolated Missing Poses, Comparisons to Same Pose and 2 Adjacent	83
5.22 CMC Curve with Pose Ground-Truth, Interpolated Missing Poses, Comparisons to Same Pose and 3 Adjacent	83
5.23 CMC Curve with Pose Ground-Truth, Interpolated Missing Poses, Comparisons to Same Pose and 4 Adjacent	84
5.24 CMC Curve with Pose Ground-Truth, Interpolated Missing Poses, Comparisons to Same Pose and 5 Adjacent	84
5.25 Example of Low Resolution Images used for Re-Identification or Appearance Model	85
5.26 Example of a High Resolution Image	85
5.27 CMC Curve with Pose Ground-Truth, Interpolated Missing Poses, Comparisons to Same Pose and 3 Adjacent with High-Resolution Images Only	86
5.28 CMC Curve with Pose Ground-Truth, Interpolated Missing Poses, Comparisons to Same Pose and 3 Adjacent with High-Resolution Models Only	86
5.29 Appearance Model Decision Tree	88
5.30 Precision, Accuracy and Recall for Threshold Values	89
5.31 First Rank Probability of Re-Identification Ignoring Matches Above Threshold	89

List of Tables

4.1	Rank Values for SIFT Features	42
4.2	Rank Values for SURF Features	42
4.3	1 st Rank Results for Tested Color Features	43
4.4	1 st Rank Results for Tested Texture Features	43
4.5	1 st Rank Results for Tested Local Features	44
4.6	1 st Rank Results for Analyzed Features	44
5.1	Testing Overview for the 3D Model	80
5.2	Result Comparison with Single-Shot State of the Art Algorithms on the CAVIAR4REID Dataset	90
5.3	Result Comparison with Multiple-Shot State of the Art Algorithms on the CAVIAR4REID Dataset	91

Abbreviations and Symbols

AbaBoost	Adaptative Boost
AdaBoost.MH	Adaptative Boost Minimize Hamming
AdaBoost.MR	Adaptative Boost Maximize Ranking
AdaPT	Adaptive Pedestrian Tracking
CAMShift	Continuously Adapted Mean Shift
CAVIAR	Context Aware Vision using Image-based Active Recognition
CAVIAR4REID	Context Aware Vision using Image-based Active Recognition for Re-identification
CCTV	Closed-Circuit Television
CMC	Cumulative Matching Characteristic
EOH	Edge Orientation Histograms
FAST	Features from Accelerated Segment Test
HOG	Histogram of Oriented Gradients
HSV	Hue, Saturation, Value Color Space
ID	Identification
KNN	K-Nearest Neighbors
LAB	Lightness (L) and Color (AB) Color Space
LFDA	Local Fisher Discriminant Analysis
LBP	Local Binary Patterns
MCTS	Multiple Camera Tracking Scenario
MIL	Multiple Instance Learning
MILBoost	Multiple Instance Learning Boost
MOG	Mixture of Gaussians
MSCR	Maximally Stable Color Regions
nAUC	Normalized Area Under Curve
PCA	Principal Component Analysis
PCCA	Pairwise Constrained Component Analysis
PETS	Performance Evaluation of Tracking and Surveillance
POV	Point of View
RGB	Red, Green, Blue Color Space
RHSP	Recurrent High-Structured Patches
SIFT	Scale-Invariant Feature Transform
SURF	Speeded-Up Robust Features
SVM	Support Vector Machine
ViBe	Visual Background Extractor
ViSE	Visual Search Engine
YUV	Luma (Y') and Two Chrominance (UV) Color Space

Chapter 1

Introduction

1.1 Contextualization

The demand for security has never been bigger. In the UK, according to a study from 2011, there is one CCTV camera for every 32 people [1] and in the US, in 2008, over 30 million cameras had been installed and used for surveillance purposes [2]. With thousands of cameras to monitor, it becomes impossible to have humans monitoring every individual. Even when a manual operator is following a single individual over a series of cameras without any prior knowledge of their disposition it's easy to lose the person. Since these sensors now capture a wide field of view with good resolution [3], there is an increasing demand for faster algorithms that can process the captured information. Currently most of this information is either discarded or manually treated instead of being automated because of the insufficient performance of current algorithms. Smart automatic video surveillance systems would detect important events and skip trivial ones. Offline visualization of content is also a laborious task that is very time consuming and could be done much more efficiently with the aid of machines. An effort has been made in trying to automatically extract important information from video content, with an increasing number of algorithms being created each year. With the technological advances, more processing power is available and more complex algorithms can be used.

Even in a single camera environment, the entire process of detecting persons, following them through a video sequence and characterizing the persons is a challenge on its own. In a multi-camera tracking system, the challenge increases since a multitude of events can change the way the system interprets the information. In fact, our perception of the world is influenced by our expectations and by all the experiences learned in a lifetime. Even though advancements have been made regarding machine-learning [4], it is still impossible to endow machines with the ability of perceiving reality the same way a human would. Several problems are added to the problems that already existed in a single camera system: the person can appear in a different pose; changes in the illumination due to the camera position affect the way image is extracted; and cameras extract information differently.

With the ability to track people in multi-camera environments, determining the route of a

criminal would be easier, control of access to unwanted spaces would be automated and suspicious behavior could be automatically detected. However, currently installed systems are still very ineffective in part because of the inability to correctly identify a particular individual in a set of cameras. For a multi-camera tracking system, a proper “object-handover” process is required, in which the identity of a tracked object is maintained when it’s detected in a new camera. This means that when an object is detected in two cameras at the same time or transitions from one camera to another (either immediately or after a period of time), the system attributes the same identity to both objects. In order to do this, the system must extract a set of distinguishable characteristics that uniquely identify each person.

One of the greatest current challenges is to define a set of features (or characteristics) that can be used to properly identify a specific person, which should work even in multi-camera systems, where calibration, synchronization, differences in resolution and color occur.

1.2 Objectives

A multi-camera tracking system is typically composed of several modules, such as foreground segmentation, people detection, people tracking and people correspondence, each with their unique challenges.

The main focus on this thesis is on people correspondence, which includes both appearance models and re-identification solutions. With that in mind, the following objectives for the thesis are defined: (1) study state of the art algorithms that can be used for extracting characteristics and matching models; (2) analyze state of the art features and techniques to evaluate their individual performance; (3) analyze the possibility of combining the best selection of features to create a more complex model; (4) expand the created model with additional features.

1.3 Contributions

Two main contributions come from this thesis: (1) the use of a multi-resolution model, which handles the problem of model creation and re-identifications depending on the available resolutions; (2) the use of a 3D model for the person capable of interpolating the missing poses.

1.4 Document Outline

This document consists of five additional chapters: Chapter 2 overviews existing solutions on the problem and important algorithms that have already been developed; Chapter 3 presents some available datasets and metrics; Chapter 4 presents and analyzes relevant features that can be used to model the person appearance when applied to a re-identification challenge; Chapter 5 details the proposed solution, which includes how these features work together to achieve better overall results and several improvements to the model; Chapter 6 presents a final discussion and future work.

Chapter 2

State of the Art

This chapter provides an overview of research in both People Appearance Models and People Tracking in Multi-Camera Environments. Subchapter 2.1 starts by introducing some concepts and provides a block-based algorithm for multi-camera people tracking. Subchapter 2.2 presents some classification methods used in several algorithms. Subchapter 2.3 includes state of the art methods for people appearance models. Subchapter 2.4 includes methods for people tracking in multi-camera environments, which start by identifying algorithms for foreground segmentation, then uses people detection algorithms to find the regions of interest and then uses single camera tracking to follow them in the scene.

2.1 Concepts and System View

When introduced, CCTV cameras were connected to display monitors and would provide a live feed to human operators. In some cases, they would also record the footage. With advances in technology, it was possible to introduce some video processing, which greatly improved the effectiveness and productivity of those human operators. This processing would include techniques to detect people, vehicles, dangerous objects, etc., and track them through the video. The ultimate goal is to be able to have an autonomous system that will process the video in real-time and detect and track objects of interest.

The process of tracking an object corresponds to the ability of following the same object through the system, determining its trajectory. In order to do this, the object needs to have an identity that is maintained over the whole sequence, which means that it must be represented by a set of features (or characteristics), grouped inside what's called an appearance model. When an object is detected, this detection is compared to the known persons in the scene to determine if it's a new person or a re-identification, which can happen within the same scene or from one camera to the other.

When speaking of multi-camera tracking systems, there are two focal aspects: (1) for each of the cameras in the system, the system needs to identify and track the object; (2) the system needs to recognize it in different cameras.

The process of tracking people over multiple cameras starts with each camera being able to recognize the objects of interest in the scene and including them in the foreground. Then, each person should be associated with an appearance model. After the initial detection is made, object tracking algorithms can be used to follow the object in the scene. A single camera block diagram can be seen at Figure 2.1, which starts with foreground segmentation (which may not be used in some algorithms), blob generation and the tracking block.

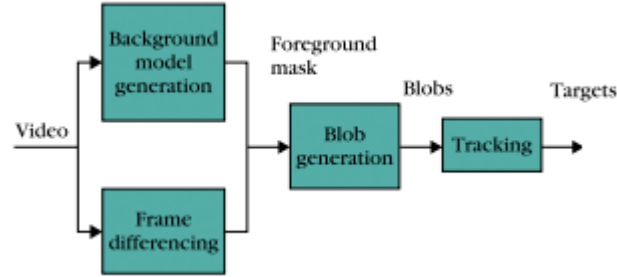


Figure 2.1: Single Camera Tracking System (adapted from [5])

The multi-camera section starts when all the detections from multiple cameras are analyzed to deal with object-handover, checking for repeated persons and labeling them accordingly. One approach to re-identify an object is to share the appearance model in between cameras. The corresponding block-diagram can be seen at Figure 2.2, which starts with an individual content analysis (each of the camera's single tracking systems) and uses data fusion to generate the final output.

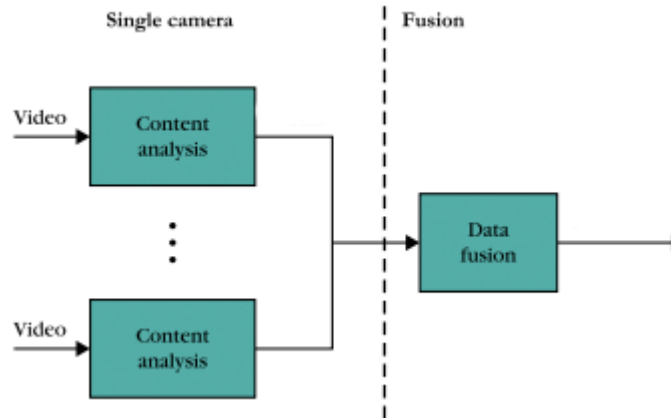


Figure 2.2: Multiple Camera Tracking System (adapted from [5])

Choosing the right algorithms for each of the blocks will directly affect the overall performance of the system, as failing in the initial steps will introduce errors in the system. However, these blocks can be tested and optimized individually as their inputs and outputs can be well defined.

2.2 Classification

Several algorithms for people detection and re-identification make use of classifiers. A classifier is, generically, a function that takes an input and generates an output. Using good extracted features, a classifier can determine if a person is present in a region [6], if a detection corresponds to a previously detected person [7] or even the person's pose [8]. Classifiers are powerful tools that when correctly applied lead to very interesting results.

The K-Nearest Neighbors (KNN) algorithm [9] uses a distance function to determine the closest matches. In the training stages, each N-dimensional input is placed in the hyper-space, tagged with the expected output value. When a new point is introduced, the distance function is applied to determine the distance between the training points and the testing point. The output of each of the K lowest distances is extracted and a voting system is used to find the best match. This is a very simple method for classification, but suffers from dimensionality problems: since all data must be kept in memory for comparison and the number of comparisons increases exponentially with the amount of data to compare, when more features are added, the computation cost will greatly increase. Improvements on the KNN algorithm have been proposed over the years, such as FLANN (Fast Library of Approximate Nearest Neighbor) [10], which chooses the most adequate search algorithm considering the available data and the wanted precision. This choice of algorithms reduces the impact of the dimensionality problem and feature size.

Support Vector Machines (SVM) [11] linearly separate two classes using training data. The input is placed in the N dimensional hyper-plane (along with the expected output, or class) and the training process will find the optimal hyperplane that will split the two classes. When a new feature vector is used for classification, the classification function will use the defined boundary to determine which class it belongs to. A two dimensional example can be seen at Figure 2.3, in which two classes (circles and crosses) are placed in a two dimensional plane and the optimal boundary, which maximizes the margin between the boundary and each of the closest training samples, is shown.

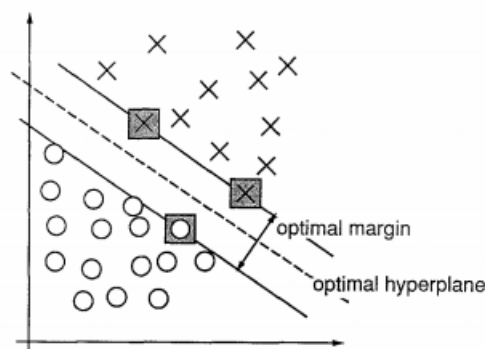


Figure 2.3: Support Vector Machines example for a two dimensional space (extracted from [11])

The way it was initially defined, Support Vector Machines would only allow separation between two classes. In some cases, this limitation is not important, but systems often include more

classes and because of that, multi-class SVM methods were introduced [12]. In this case, a one-vs-all approach is used, which transforms what would be a multiple dimension optimization problem to a single dimension optimization problem. Another more computationally heavy approach is the use of several one-vs-one Support Vector Machines, which according to a comparison presented in [13] shows better results, but with higher computational cost.

Boosting algorithms iteratively add new weak classifiers to make a strong classifier. A weak classifier separates two classes at least as well as a random classifier (*i.e.* with at least 50% success). According to their performance, they are weighted to construct the strong classifier. One of the most commonly used boosting algorithms is AdaBoost [14], which is simple to implement and is very good at selecting the most representative features. To add support for multiple classes, some extensions have been proposed, which include the Adaboost.MH and Adaboost.MR [15] that extend Adaboost to a multi-class, multi-label decision. More recently, another proposed extension to the Adaboost algorithm introduces a decision tree [16], which greatly improves the processing time without a significant loss of performance.

Recently, fuzzy classifiers have gained a lot of focus in investigation. They are classifiers that use fuzzy sets or fuzzy logic [17], which introduce uncertainty in the model. This means that while data can be trained from a set, the algorithm is capable of self learning the additional, untrained cases.

While the application to the K-Nearest Neighbors algorithm [18] or Support Vector Machines [19] aren't particularly recent, their use as detection or recognition algorithms has recently increased thanks to the advancements in shape extraction methods, such as [20]. While interesting, further analysis of these classifiers falls outside the scope of this thesis.

2.3 Appearance Models

One of the ways of establishing object correspondences between cameras is by using the object's appearance. The objective is to choose a feature or a set of features that provide a discriminative visual signature, fully representing the object. Several approaches can be made to these appearance models depending on the situation: a greater array of features can be used to improve detection but will also increase computation and memory costs that may make it unsuitable for a real-time system. Another important part of the structure of the appearance model is way the system determines how close an input is to a model, in order to establish a match.

2.3.1 Types of Features

2.3.1.1 Color Features

Some approaches use color information, such as in [21], which uses the object's color histograms to model the appearance. It's considered that the color information is affected by Gaussian noise to compensate some of the changes that occur due to noise and small illumination changes. In a multi-camera network, there is also the need to consider that cameras may be different and may be using different sensors which will capture the images differently. To use this type of information to make the correspondence, a process of color calibration can be made (also called "Colorimetric Calibration"). When the cameras have a similar field of view, this can be done with the "Brightness Transfer Function", introduced in [22], which uses both a distance metric and known matches between the cameras (such as recognizable objects), analyses the color differences and builds a model that represents the transformation, with the result seen in Figure 2.4. An extension of the algorithm is proposed at [21, 23], which learns the Brightness Transfer Function for each pair of images in which an object is detected, and determines a final function using Principal Component Analysis (PCA) [24] on all the available functions. The limitation of these algorithms is that they require the same objects to be matched in a similar point of view (POV), limiting the scene structure. In [25], the algorithm starts by accumulating images of the same tracked object in each camera before applying the Brightness Transfer Function, creating a better probability of finding matches.

Also using color information, the Visual Search Engine (ViSE) [26] segments the person in three parts: head, torso and legs, maintaining a spacial correlation between them (the head should be above the others, connected to the torso below, connected to the legs further below). The algorithm ignores the head portion for making correspondences as the authors consider that it's hard to make a reliable match with it, focusing on the torso and legs and describing them in an HSV color histogram. The histogram uses 10 bins, which the authors considered to represent the most distinctive colors detected by the human visual system. Correspondences are made with the torso and legs, verifying that their most common value is the same as the one in the appearance model.

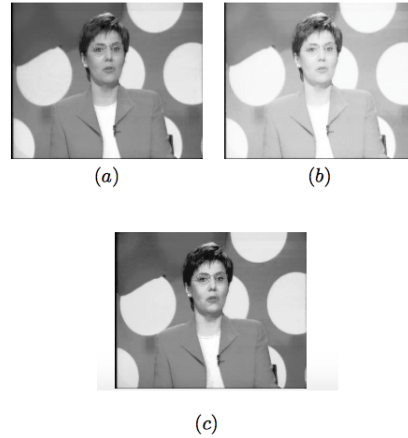


Figure 2.4: Example of Application of the Brightness Transfer Function (extracted from [22]). (a) Reference; (b) Over-exposed Image c) Image after applying the Brightness Transfer Function

The use of different body parts can be taken advantage of as in [27], where the person is divided into 6 parts: chest, head, thighs and legs, modeled individually using an HSV histogram. The features are then concatenated into a single feature vector and information on the black color is added to deal with the problems of low saturation and brightness. The algorithm is then extended to a multi-shot model which uses multiple images to add robustness to the model.

The authors of [28] extract model information from the HSV and YUV color spaces and create a feature vector with 11253 dimensions. It then has its dimensionality reduced using the unsupervised PCA, followed by the Local Fisher Discriminant Analysis (LFDA) [29] to optimize the distance function for comparison.

In [30], the person image is split into 25 overlapping rectangular segments. For each, three RGB histograms are built, one for each channel, with 4 bins each. Images are compared using the Hellinger distance metric, in which 0 represents complete similarity. The chosen structure is the hierarchical feature-distribution scheme that allows finding the best match in a faster, more efficient way. The authors also apply rules for forgetting older models to avoid an uncontrolled increase in the number of comparisons and persons who have long left the scene and are unlikely to return.

2.3.1.2 Local Features

Local features extract information from a series of keypoints in the region of interest. One method, [31], uses an accumulation of Speeded-Up Robust Features (SURF) [32], which are extracted over multiple detections of the same person. The points are stored in a KD-tree model for speed purposes and when a person needs identification, the features are extracted and a voting system will find the best match with similarity being evaluated by the Sum of Absolute Differences.

A comprehensive test using local features was presented in [33], which combined a wide variety of different feature detectors with feature extractors. They then create a bag of features to establish the structure of the model and normalize the distance for comparison purposes. A

combination of Gradient Location and Orientation Histogram (GLOH), from [34], and Scale-Invariant Feature Transform (SIFT), from [35], have shown the best results.

2.3.1.3 Mixed Features

In [36], Support Vector Machines (SVMs) [11] are used to classify color-based and shape-based features. The extracted color features are normalized color histograms, while the extracted shape features include a shape histogram of 3 by 3 matrices similar to edge detection, as seen in Figure 2.5. The authors come to the conclusion that using two dimensional normalized color histograms has better results.

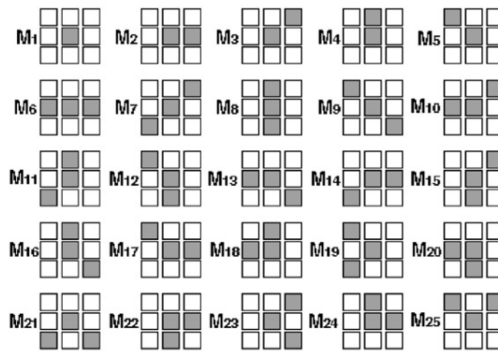


Figure 2.5: Extracted Shape Patterns (from [36])

Another approach takes advantage of the use of templates [37] by modeling humans as a puppet of rectangles, as seen in Figure 2.6, and uses those different rectangles as individual templates. The authors compare a bottom-up approach (which looks for candidate body parts in the frame) with a top-down approach (which looks for the entire person in the frame). The method they present first detects candidate parts with an edge detector, cluster the patches to identify body parts and prune clusters that move too fast to filter out the unwanted patches.

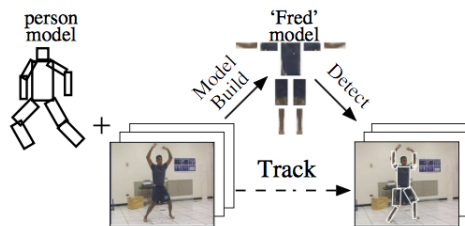


Figure 2.6: Puppet-like Representation of the Person (extracted from [37])

In [38], the Canny Edge Detector [39] is used to detect edges, and along those edges several patches are extracted with position (distance and angle from the patch to the head) and color. Since the distance function could prove to be ineffective when the person came closer or further away from the camera, this information is normalized to the person's height. When a new person is identified, an algorithm will extract the patches and use a distance function to compare to all

stored models. If the result is above a threshold, it's a new person, otherwise it re-identifies as the result with the lowest result.

In [40], the authors start by dividing the image window in overlapping blocks. Texture information is extracted using co-occurrence matrices, which represent second order texture information [41]; edge information is extracted using the Histogram of Oriented Gradients method; color information is extracted through normalized color histograms. The feature vectors then have their dimensionality reduced using the Partial Least Squares method, which is done by projecting the feature vector in the desired dimensional space. Using an all-vs-one approach, the Euclidean distance is used to compare new feature vectors to those of previously detected persons.

The Symmetry-Driven Accumulation of Local Features method [42] starts by finding the axes of asymmetry and symmetry for the pedestrian, assuming that only the foreground is present; then, using the symmetry information, the algorithm extracts weighted color HSV histograms, the Maximally Stable Color Regions (MSCR) [43] and texture information using Recurrent High-Structured Patches (RHSP) [42]; finally, re-identification is done using a matching distance, which gives different weights to these components to optimize the results.

Some authors used learning models, such as in [44], which calculates an optimal distance function from the extracted color and texture features. Each person was represented by a feature vector of 2784 dimensions, with a distance function being defined by using a testing set and maximizing the re-identification rate.

The approach in [45] uses both global and local features to extract the relevant information from the person. The algorithm starts by separating the foreground and the background to remove unwanted noise. Then, an asymmetry segmentation is done to extract what the authors consider to be the most relevant parts of the pedestrian from which features are extracted. This segmentation will divide the detected person horizontally to obtain the head, torso and legs. Then, an HSV histogram is used as the global feature. When multiple input models are available, this histogram represents the average of the individual histograms to add robustness to changes in illumination and pose. The authors also create a collapsed mode made from overlapped patches to obtain an even model.

In [46], both color and texture information are stored to try to overcome the changes in the point of view a person can appear. First, the algorithm divides the detected person in overlapping patches. In each of the patches, the mean values per color channel are calculated and discretized, using the HSV and Lab color space models. As for texture information, it is extracted using Local Binary Patterns [47]. Color and texture information are stored in a feature vector. All feature vectors are concatenated to have a representation of the whole image and PCA is used to reduce the vector dimension. The result is then used in a classification algorithm which, in this case, is the Large Margin Nearest Neighbor [48] classifier. This machine learning approach is usually done offline and requires a set of positive and negative matches to train properly.

Another solution that combines color and texture is proposed in [49], which extracts color histograms in 3 color spaces (RGB, HSV and YCrCb) as well as Local Binary Patterns to create the feature vector. Taking pairs of matches, their algorithm learns the optimal distance function to

use in future comparisons. The main advantage of the Pairwise Constrained Component Analysis (PCCA) is the ability to handle high dimensional data.

Random-Projection-Based Random Forest are used in [50] to extract the relevant features from the 2592 dimension feature vector from color and texture features. The classifier is trained with a subset of the testing dataset. The main advantage of the method is that when compared to other learning models, their solution is faster and with better re-identification rates.

2.3.2 Model Approaches

Appearance models can be defined as single-shot models or multiple-shot models, depending on how many instances of the person are stored. Single-shot models are usually faster since there's only a single model for the person to compare to, like [26] or [38]. Other approaches store multiple versions of the model, such as in [31]. While multi-shot models usually perform better, some systems may have time or memory constraints that forbid this approach. An extension of the multiple-shot approach is the use of a 3D Model. In [51], the authors use information from different frames, in which the person is captured in different perspectives, to create a 3D body model. The orientation of the person is estimated based on [52]. The plane image is then projected to the 3D model according to that orientation and the features are extracted according to a series of predefined vertexes. The model is updated as more information is retrieved. A distance function evaluates if a detected person can be matched to an existing 3D model. The main advantage of this type of approach is that it allows to do re-identification even if the person is captured from a different angle.

Another difference in the approach is the use of learning models. Some algorithms choose to use learning models to determine the best distance function, such as in [44, 46, 49]. However, this approach can often lead to over-fitting, which means that the algorithm would need to be trained for specific datasets and scenarios instead of being a more generic solution.

2.3.3 Object Correspondence

When trying to track objects, it's important to correctly re-identify objects on consecutive frames. The process of object correspondence is to find a match for the same object, which is also commonly referred to as object re-identification. Going from a single camera to a multiple camera network, challenges such as differences in angles, distances to camera or illumination add complexity to the problem. Transition correspondence is based on finding relationships between objects exiting and entering the cameras' field of view, linking entrance and exiting zones and finding relations in between: if an object disappears from a certain camera it's likely it'll reappear on another area/camera [53].

Statistical models have been used to determine the probability of matching an object that disappears from one camera to a specific part of another camera [53]. This is done by calculating the mutual information, entropy and posterior expectation and use that information to create a probabilistic model.

Gaussian Models have been used to represent entry and exit zones, which give spacial and probabilistic information [54]. Since people usually follow common paths, there are specific points where the object detection will end. Clustering these points with K-Means or Maximum Expectation algorithm detects the points where entries and exits more often occur. Using the information in [54], the addition of temporal correlations between the entry and exiting of objects improves the estimation as it predicts when the object is expected to appear [55]. A probabilistic approach, when compared to a deterministic approach, can work with non-linear inter-connection paths and will cope better with multiple hypothesis without using computationally expensive methods such as particle filters. Some examples of results using Gaussian Models are shown in Figure 2.7.

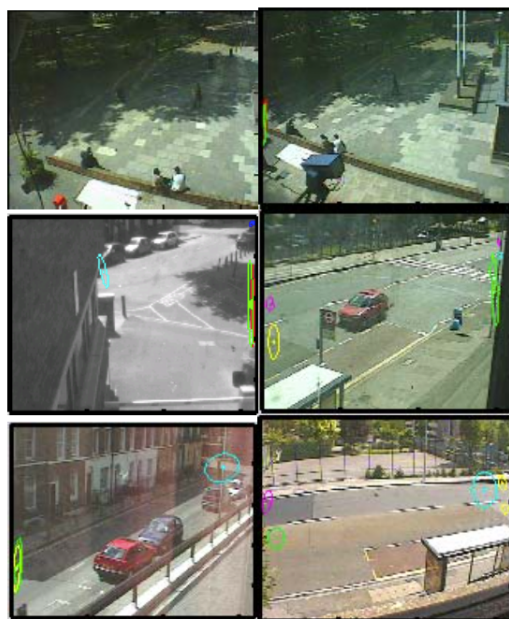


Figure 2.7: Example of the use of Gaussian Models to find and Match Entry and Exit Points from the scene (extracted from [55])

A complete model, from building a complex network of cameras, matching the appearance and modeling the information in a statistical model as well as validating the topology by using the mutual information model is proposed in [56]. It uses a graph model of the cameras in the scene to establish common camera transitions. This way, on a complex system, the probability of objects transitioning from one camera to another can be defined and used when searching.

2.4 People Tracking in Multi-Camera Environments

The process of tracking people in Multi-Camera Environments is usually divided into different stages: pre-processing, where the noise is removed, which can be done using foreground segmentation techniques; detection, where the scene is analyzed and objects of interest are detected; object tracking, where a previously detected object is followed.

2.4.1 Foreground Segmentation

This section will focus on techniques that allow separating the background from the foreground on a video sequence. The foreground can be defined as the objects of interest in the scene. If the background is known *a priori*, simply verifying the differences to the model would be enough. However, some challenges, such as illumination changes, irrelevant motion (for example, wind blowing the leaves of a plant when trying to track people) and occlusions can wrongfully classify a portion of the scene. A bad foreground detection can lead to poor results since it's commonly used as the first step in the processing pipeline. The foreground segmentation models follow a generic structure seen in Figure 2.8, which starts by initializing the background, using the information on new frames to improve the background model and using this model to detect the foreground mask.

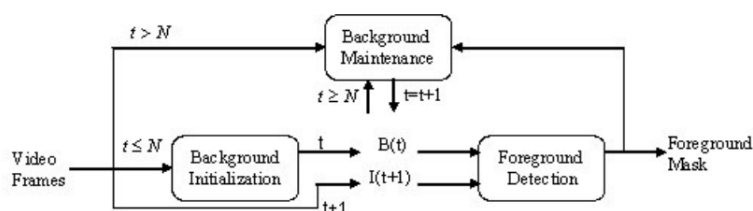


Figure 2.8: Background Subtraction Process with N frames for background initialization and B and I being the background and frame images (extracted from [57])

Frame differencing [58] introduced the possibility of using foreground segmentation algorithms in real time systems. In this method, objects of interest are considered to be the difference between two frames, on a pixel by pixel basis. If the difference between a pixel in a frame and the same pixel in the previous frame is greater than a threshold, then that pixel is considered to be part of the foreground. This method provided a computationally inexpensive method of getting the foreground information. However, it's very dependent on a good threshold value and will be unable to deal with compression artifacts and changes in the scene illumination. Since it is a very fast method, some additional processing can be done to the images to eliminate some disturbances, such as using more complex thresholding operations or morphological operators [59, 60]. An example of the use of frame differencing can be seen in Figure 2.9, in which the reference frame is placed 3 frames before the current one. When subtracting the current frame with that reference, the result is seen in (c), and is then thresholded so that only the relevant differences are considered (d). Additional processing can be done, such as using morphological operators like erosion to

remove unwanted areas (e). Finally, the center of gravity of the target region is used as the output (f).



Figure 2.9: Application of the Frame Differencing method (extracted from [60])

Background Subtraction is an evolution of the frame differencing method, using a more complex Background Model instead of a previous frame. In the Mixture of Gaussians (MOG) method, each pixel is modeled by several Gaussian Curves. When a new frame is introduced, each Gaussian from each pixel is updated with the new information [61]. This method, while computationally more expensive, proved to be very robust, even in outdoor scenes (provided that the illumination changes weren't too significant) and is widely used for foreground segmentation. To deal with illumination changes, some modifications were proposed, such as in [62], that combines intensity based representations with color based representations, determining that the changes that only occur in the intensity model but not on the color model are probably illumination changes and don't represent foreground objects. Several other methods have been proposed with variations on the Mixture of Gaussians method [63], including the ability to handle noisy images, camera jitter and changes to the background itself.

Shadows have been a problem in the detection of the foreground as they are not part of the background model but it's not desirable that they are detected as objects of interest. A proposed generic solution used a pixel-based deterministic model based on the HSV (Hue, Saturation, Value) color space [64]. It classifies pixels as shadows when: (1) Hue and Saturation are sufficiently close to the background model's values; (2) the ratio between the Value and background value

falls between a pre-defined range. Since all parameters can be changed, they can be tuned for the type of scene under analysis. Another method uses the RGB (Red, Green, Blue) color space [65], in which the authors consider shadows to be offsets from the background model, applied to each of the channels. They use different thresholds to determine if the change is due to a shadow, noise or a different object. In [66], Bayesian Probabilities are used to model both the background and the shadow, determining the probabilities of certain values in the YUV color space being shadow, background or foreground. Comparing the pixel value with the background model's value, a probability for the pixel being a shadow is found. The authors also use a similar method, but applied to finding foreground pixels, to improve the model and segmentation results. A comprehensive survey on shadow detection methods has been published in [67], which includes methods that use color information, geometric information or texture information.

In [68, 69], ViBe, a universal background subtraction algorithm is proposed which aims to build a dynamic background model. Each pixel in the background model is represented by twenty samples. On a new frame, N random samples are extracted to update the model. When a background value is selected, the neighbor pixels are also used, adding a spacial component to the algorithm. A pixel is said to belong to the background if the Euclidean distance to at least two samples for the pixel is below a threshold. Old and new samples have the same weight in the system and can remain in the system an indefinite amount of time. An example of application of the ViBe can be seen in Figure 2.10. The algorithm has shown very good results and several changes have been proposed to further improve it. One important change to the ViBe technique is the addition of a shadow removal algorithm [70], based on the ideas presented by [64], which assumes that a shadow corresponds to a less bright background area and evaluates the changes in Hue, Saturation and Value when comparing the frame with the background model. Also based on the ViBe algorithms, other changes were proposed [71], most notably: not allowing the use of detected foreground pixels to be part of the background samples; removing small blobs on the segmentation mask; filling holes in small areas of the update mask; limiting the propagation when the gradient value is high; changing the distance metric from Euclidean distance to a color distortion metric based on the work presented in [72]; adding an heuristic to detect pixels that change between background and foreground (called blinking pixel). These changes have shown several improvements in the testing scenarios.

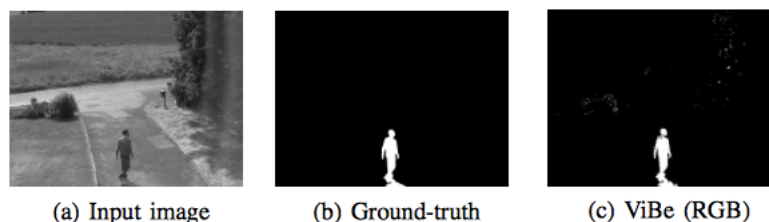


Figure 2.10: Example of a result from the ViBe Method (extracted from [69])

2.4.2 People Detection

After removing the unwanted background information, methods for people detection can be used. Using certain identifiable characteristics, such as body shape or texture, it is possible to detect a person in the scene.

One of the first approaches for people detection used Haar Wavelets [73]. These functions result in the multiplication of neighboring pixels with a series of weights. The choice of these weights can maximize specific shapes in the scene, including humans. Haar-like features are still commonly used for face detection [74].

Initially used for face detection, Edge Orientation Histograms (EOH) [75] store histograms of the orientation of edges and use that histogram information as features for a classifier. While it was not initially applied to people detection, they were then combined with Haar-like features and used in [76] to achieve similar results when compared to the Histogram of Oriented Gradients (HOG) descriptor. The authors use the Adaboost [14] classifier to achieve fast results even when using two sets of features.

The Histogram of Oriented Gradients is a feature descriptor used for object detection, commonly applied to people detection [77]. It describes the appearance of an object using the gradients in different directions. A picture is divided in cells and the algorithm calculates the gradients' directions for the pixels. It then stores the information in an histogram, which is used as a descriptor. The decision is then made using a Support Vector Machine (SVM) [11] classifier. The method shows great results when detecting humans and has been used for that purpose regularly. A variation of the HOG algorithm [78] uses a cascade of stage classifiers and takes advantage of Adaboost to build a fast method for people detection. Cascade classifiers reduce the processing time since negative samples are detected in early stages. The authors show that they are able to achieve fast processing times and very high success rates with the method.

Local Binary Patterns (LBP) are based on the Texture Spectrum model [47], which has been widely used for people detection, particularly for face recognition [79]. The creation of an LBP feature vector starts by splitting the frame into cells. For each of the pixels in the frame, the closest neighbors are analyzed along a circle. If the pixel value is above the center pixel, then a "0" is written, otherwise, a "1" is introduced. This means that an 8 bit word is formed, as seen in Figure 2.11. Then, an histogram of occurrences in the cell is created and normalized. All histograms are concatenated to create the feature vector used in the classifier.

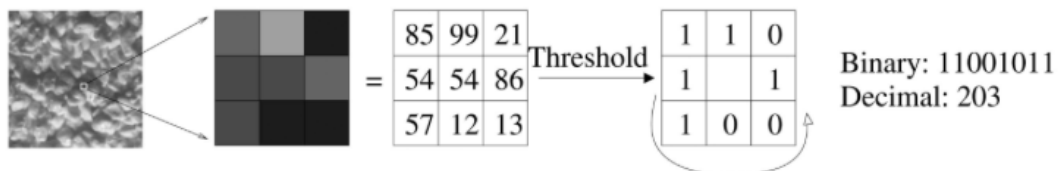


Figure 2.11: Example of Application of the LBP Method on a Patch (extracted from [79])

In order to take advantage of both methods at once, a HOG-LBP solution has been employed

in [80]. Two steps occur simultaneously: (1) the LBP is computed at each pixel; (2) a gradient is computed at each pixel, then it's convoluted with a trilinear interpolation. The vectors are merged and used as feature vectors for classification with an SVM being used as the classifier. Using both algorithms together, a better result is achieved when compared to using them separately.

Other algorithms use different combinations of features and classifiers for human detection. One example is the algorithm proposed in [81], which chooses a very specific set of Haar Wavelets for pedestrian detection to extract features and uses an Adaboost Tree Classifier [16] to optimize the speed of the algorithm. Their Haar Wavelets are applied with different sizes to successfully detect over multiple scales.

A common issue when detecting people is occlusions. When people are partially occluded, some detections will fail as they rely on a full body representation. The HOG-LBP method [80] was already able to handle with some partial occlusions, but more recent methods have improved upon it. In [82], partial detections are made as the algorithm not only detects parts of the human body (instead of trying to find the full body), but uses a probabilistic approach to estimate the likelihood of a body part being visible. The general framework is presented in Figure 2.12, which starts with partial detections, which are ranked in fidelity, and uses that information to obtain the visibility probability. If a match is good enough and is likely visible, it is placed in the frame.

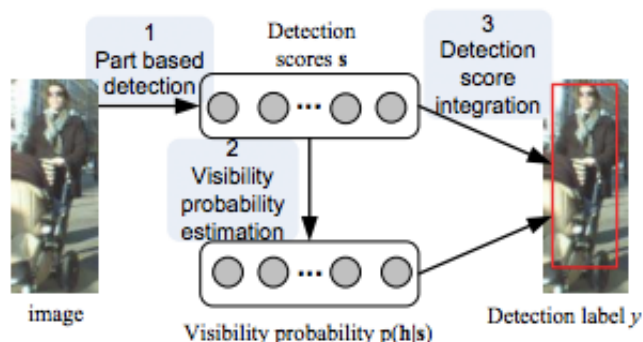


Figure 2.12: People Detection Framework with Visibility Probabilities (extracted from [82])

Other people detection methods can use 3D models. In [83], a three-stage process is proposed to determine the 3D pose of a person using a 2D frame. The first step estimates the 2D articulation and directionality of the person from a single frame. The data is then associated between frames to accumulate the 2D data from several detections to have a more robust estimation of the 2D limb positions. Finally, the 3D pose is then estimated using a Maximum-a-posteriori Estimation.

2.4.3 Object Tracking

Object tracking corresponds to the process of following a moving object over time. It's possible to use the information from previous frames to improve the effectiveness of searching for a particular object as a temporal correlation is maintained in a video sequence.

The Mean Shift algorithm is an iterative object search algorithm, first proposed by Fukunaga and Hostetler [84] that started being used after the publication of [85], which generalized the method to non-flat Kernels. Initially, the information of the object is extracted and the window size and position are chosen. In the next frame, a center of mass of the wanted features is calculated. This center of mass locates the position, in the window, where the wanted characteristics are centered. The window is then moved to the center of mass location (the "mean" location) and these steps are repeated until convergence or when the location movement is less than a threshold. It's a fast tracking algorithm that can be used in real-time applications for non-rigid objects that are well defined by color and/or texture. We can also use an updatable back projection of the object to compensate for small changes and improve the tracking results [86].

The CAMShift algorithm uses the Mean Shift algorithm to find the window center and makes size and rotation adjustments to find the best match [87]. The algorithm starts by having the initial window location chosen. Then, the Mean Shift algorithm is applied and the zeroth moment (which can either be area or size) is stored. The window is adjusted according to the zeroth moment for the next video frame with the scale and rotation of the window being updated. This allows the algorithm to adapt itself to the structure of the data, hence the name Continuously Adapted Mean Shift.

The Kalman Filter, proposed in [88], is a widely used mathematical method with many applications, including object tracking. It's a two step method: (1) prediction stage, which uses *a priori* knowledge to estimate the new object position; (2) update stage, in which the information of the new observation is used to update the model. When applied to object tracking, an assumption is usually made when modeling the objects: constant velocity. This means that the object's velocity is calculated and updated, and new estimations are made assuming that the object keeps a linear trajectory, which is one of the main limitations of the Kalman filter. To overcome this, an extended Kalman filter has been proposed [89], using the Unscented Transform to apply non-linear transformations, which allow different types of non-linear movements. One setback of the Kalman filter is that it is a single hypothesis filter and because of that, recovering from incorrect readings may prove troublesome.

Particle Filters, also known as Monte Carlo methods [90], approximate a probability distribution using particles. Unlike Kalman filters, multiple hypothesis are held which add flexibility but also another layer of difficulty in determining parameters such as intended density or number of particles. These particles start by representing points in the initial model, which are chosen randomly. Then, for each of the points, non-linear equations are applied to estimate the next state and the observation will adjust the results. One advantage of the method is that the estimation error will converge to 0 as the number of particles increases. However, using a large number of

particles will also increase computation cost. Some extensions have also been proposed to Particle Filters, such as adding Adaboost [91] or a method such as mixture tracking [92] which allows the algorithm to deal with multiple targets.

In [93], a 3D People Tracking method has been proposed, which builds a probabilistic tracking model of the person. When the person is visible, the model is updated with the new information, but the tracker is still able to predict human motion even when the person is occluded. The model is built so that it stores the walking cycles of the several persons, and models them in a probabilistic distribution to predict the movement.

Some modifications to tracking algorithms are made to support the notion of 3D environments, such as in [94], which uses a geometric consistency analysis. The initial tracking is made on the head of the person and the final position is detected from it. The algorithm receives the 2D tracking information from each of the cameras and computes the Error Measurement by using the number of cameras as a parameter, the Euclidean distance from the head to the camera plane and the projection of the 3D position to the camera plane. This method requires previous camera calibration, but provides good results in cases of occlusion.

The Real Time Adaptive Pedestrian Tracking (AdaPT) [95] uses a combination of both deterministic and probabilistic trackers. A confidence value associated to each of the trackers to choose the most adequate one for the frame. A motion model is also built for each of the pedestrians which will estimate the velocity of each subject in the scene. That way, confidence value uses both low level data (such as each tracker's information) as well as high level data (using the generated model). The general framework model can be seen at Figure 2.13, where the frame information is used as input on multiple trackers. Then, a "tracker confidence" metric is calculated and the best tracker is chosen according to the confidence value. The output should represent the most accurate tracker result.

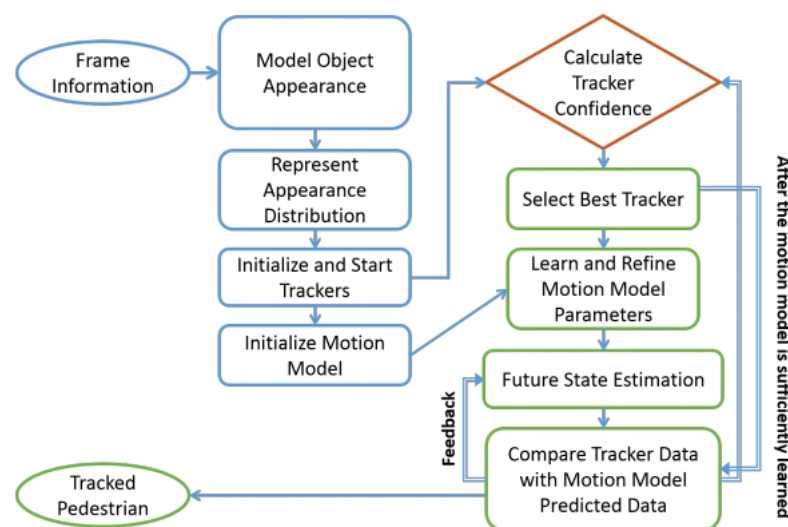


Figure 2.13: AdaPT General Framework (extracted from [95])

Another tracking system [96] focuses on finding stable bounding-boxes. It uses the head location as a starting point to estimate the bounding box. It uses the data over a sliding window of frames to minimize the errors. The algorithm is capable of handling missing observations. Since estimating the head after determining a centroid position produces more variations than starting with a head detection, this leads to a more stable bounding box over the frames.

Other proposed tracking algorithms are based on machine learning, such as [97], which uses Haar-Like features to represent the images and an appearance model based on Multiple Instance Learning (MIL) [98]. Using the probabilistic results from the appearance model, the tracker estimates the most likely position for the person. The main difference compared to the use of a regular classifier is that instead of matching feature vectors to binary data, they match a bag of features to a bag of labels, which is called Multiple Instance Learning. The authors present a boosting classifier named Online-MILBoost, which solves the MIL problem with a real time boosting algorithm.

Chapter 3

Datasets and Assessment

To properly evaluate the results, two things are needed: a metric, which quantitatively evaluates the results; and a dataset, which includes relevant video or images that are being processed, with relevancy being using content that could represent their use in a real life scenario. Failing to choose appropriate metrics leads to a poor evaluation of the results, which is often used to decide the next approaches. On the other hand, if the dataset is not adequate and representative of the situation where the system will work, the metric is evaluating unrealistic situations.

Metrics are chosen according to their objective: classic metrics, like precision and accuracy are used together with re-identification metrics to test the proposed model.

As for the datasets, they are chosen according to the availability, type of sequences to test and challenge. In this case, the objective is choosing a dataset that allows testing the re-identification rate.

3.1 Tracking Datasets

The CAVIAR Dataset [99] provides a set of videos captured from two points of view in a shopping center. It contains 26 different scenarios that allow testing the algorithms in very different situations. One big advantage of using this dataset is the availability of ground truth data, with information on identification and position of the objects. In Figure 3.1, an example of an annotated frame is shown, which includes individual information and group information as well as ground-truth data for multiple body parts. In Figure 3.2, an example of the two perspectives from the CAVIAR dataset is shown, with visible changes in color, scale and perspective. These challenges and the fact that it was extracted from a real life scenario is why it's one of the most commonly used datasets in literature.

The I-LIDS Multiple-Camera Tracking Scenario (MCTS) [100] includes sequences captured in an airport, captured from different cameras that cause severe illumination changes. While popular, this dataset is no longer accessible for free download.

While original used as a way of testing human detectors, the ETHZ [101] has gained popularity as a tracking dataset. It is composed of four sequences captured by moving cameras, which forces

big variations in the person appearance, creating an additional challenge for both tracking and testing appearance models.



Figure 3.1: Annotated frame example from the CAVIAR dataset (extracted from [99])

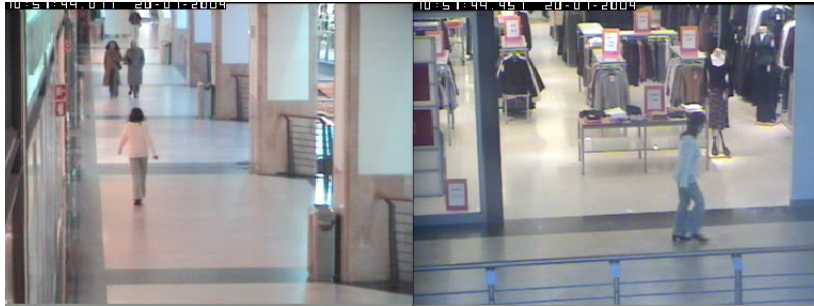


Figure 3.2: Frame example over two cameras from the CAVIAR dataset (extracted from [99])

The PETS 2001 Dataset [102] includes annotated multi-view sequences that include occlusions and changes in lighting. This dataset also provides ground-truth annotations to test the performance. In Figure 3.3 the camera perspectives available with this dataset are seen. Several camera angles are available which introduce different challenges for the algorithms.

3.2 Re-Identification Datasets

The I-LIDS Video re-IDentification (I-LIDS-VID) Dataset [103] is extracted from the I-LIDS Multiple-Camera Tracking Scenario and uses 600 images sequences from 300 individuals: two sequences, one from each camera, for each person. The sequences last from 23 to 192 frames, with an average of 73. The challenge of these sequences comes from not only the pose and illumination changes but also from the clothing similarities, background noise and occlusions. As it was the case with the I-LIDS MCTS, this dataset is no longer available for free.

Another test to the performance of the re-identification algorithm is with the VIPeR Dataset (Viewpoint Invariant Pedestrian Recognition Dataset) [104]. The objective of this dataset is to



Figure 3.3: Examples from the PETS 2001 dataset (extracted from [102])

test how viewpoint invariant the appearance models are. For this, there are 632 pedestrian pairs, with each pair consisting of the same individual in different poses, classified in 45° angles and in varying illumination conditions. Images from this dataset have been scaled to 128x48 pixels. An example of a pair of pedestrians is shown in Figure 3.4.

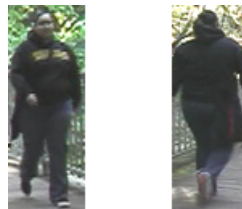


Figure 3.4: Example from the VIPeR Dataset

The VISOR dataset [105] is created for multiple-shot models as it includes 200 snapshots from 50 persons (four different views for each) to use for creating the models and short clips for testing the re-identification.

To test the performance of the people feature extraction and re-identification, the CAVIAR4REID dataset [27], extracted from the CAVIAR dataset [99], provides 10 to 20 images from 72 individuals, observed in both cameras. The dataset includes variations of the individuals: pose and resolution. In Figure 3.5, a set of example images from the CAVIAR4REID is shown. In these, only the region of interest (with background) is present.

The CAVIAR4REID Dataset will be the main dataset chosen for testing re-identifications. It includes a wide variety of images extracted from a real-life scenario, the CAVIAR sequences,



Figure 3.5: Examples from the CAVIAR4REID dataset (extracted from [27])

with different people, poses and resolutions. The CAVIAR sequences are also one of the most commonly used datasets for People Tracking algorithms. The VIPeR dataset will also be used since it's a common dataset for re-identification solutions.

3.3 Metrics

For the initial stages of creating and testing the appearance model, evaluation is made using the Normalized Area Under Curve (nAUC) metric [106], which provides a scalar measure to the re-identification performance. This metric represents the area of the Cumulative Matching Characteristic (CMC) Curve, which in turn is the expectation of finding the correct match in the top n matches. However, since the first rank is the most important factor (as it represents the probability of re-identification in a single best match), the 1st rank of the CMC will be directly used to evaluate the results. The curve is cumulative in the sense that, for example, the third rank represents the probability of having the right re-identification in any of the three best matches. On an ideal system, it would successfully re-identify at the first match (first rank). From a set of M images to re-identify and P possibilities, from p_1 to p_P , considering the correct option to be C , the CMC

value can be determined according to Equation 3.1. This means that the CMC is given by the average from all images of the correct matches where any index below k is correct.

$$CMC_k = \frac{1}{M} \sum_1^M \sum_{i=1}^k \begin{cases} 1 & p_i = C \\ 0 & p_i \neq C \end{cases} \quad (3.1)$$

In other situations, classic metrics such as precision and recall [107] can be used to evaluate the results. The definition of a true positive is made according to the test setting. The precision metric is defined in Equation 3.2 and can be interpreted as how well the algorithm works when it makes a positive detection. A perfect value of 1 means that all the positives given by the algorithm are true positives. The recall metric is defined in Equation 3.3 and is used to determine if the algorithm is detecting all the positive events that occur.

$$\frac{\sum TruePositives}{\sum TotalAlgorithmPositives} \quad (3.2)$$

$$\frac{\sum TruePositives}{\sum TotalExistingPositives} \quad (3.3)$$

Chapter 4

Individual Features

4.1 Feature Overview

Features are an important part of the appearance model. They are the elements that represent the image (or portion of the image) and are often called the signature of the image [108]. They include elements such as color, texture, both of which can be extracted on a global level or local level [109]. Additionally, features can be combined for better results.

Global features are extracted from the entire image (or region of interest) and include, for example, an histogram of a channel of the image. Local features are extracted from the neighboring area of relevant points, which adds the extra step of locating these points.

Since there are a multitude of features that could be used, the first part of the dissertation consists in finding the ones that would produce the best results. The testing procedures are done in two stages: initially, an optimization of each of the elements is made and, depending on the results, additional experiments are conducted.

The chosen features include Grayscale values and histograms, RGB histograms, Hue values, HSV histograms, CENTRIST histograms, Haar Wavelets histograms, Edge Energy value, Laplacian histograms for global features and SIFT and SURF under different keypoint detectors for local features.

4.1.1 Global Information

4.1.1.1 Color Information

Color Information is commonly used to identify objects or people [110, 111, 45]. Two types of experiments are made: (1) the amount of information extracted, which goes from a single mean value to storing a complete histogram with a varying number of bins; (2) different color spaces.

4.1.1.2 Grayscale

The grayscale color space represents the luminance, which corresponds to a representation of how our visual system perceives illumination. This means that the grayscale color space isn't the

average of the RGB channels, but a weighted average. It is defined in Equation 4.1.

$$L = 0.299 \times R + 0.587 \times G + 0.114 \times B \quad (4.1)$$

Different types of information can be extracted from the grayscale colorspace, including the mean value and histograms. For the mean value, two cases are tested: (1) the mean value of the entire region-of-interest; (2) the mean value of each of the three body parts proposed at [112], in which the person is divided into three sections: head, torso and legs. The height of each is in Equation 4.2, as defined in [112]. An example of the division is shown in Figure 4.1.

$$\begin{cases} Height_{Head} = 11.76\% \\ Height_{Torso} = 41.18\% \\ Height_{Legs} = 47.06\% \end{cases} \quad (4.2)$$

Similarly, one (for the full body) and three histograms (one for each body part) are tested under similar conditions.



Figure 4.1: Example of the 3 Body Part Division, from the CAVIAR4REID Dataset

4.1.1.3 RGB

The RGB (Red, Green, Blue) color space represents the image using three channels, one for each of the colors directly perceived by the human eye. This model is also a close representation of the way information is captured.

These features are tested using a total of nine histograms, three for each channel and one for each body part in each channel. The division in body parts is again from [112].

4.1.1.4 HSV

The HSV (Hue, Saturation, Value) represents the color, saturation and brightness of the image in three separate channels. It is defined in Equations 4.3, 4.4 and 4.5.

$$V = \max(R, G, B) \quad (4.3)$$

$$S = \begin{cases} 0 & \text{if } V = 0 \\ (V - \min(R, G, B))/V & \text{otherwise} \end{cases} \quad (4.4)$$

$$H = \begin{cases} 60 \times (G - B)/(V - \min(R, G, B)) & \text{if } V = R \\ 120 + 60 \times (B - R)/(V - \min(R, G, B)) & \text{if } V = G \\ 240 + 60 \times (R - G)/(V - \min(R, G, B)) & \text{if } V = B \end{cases} \quad (4.5)$$

The clothing color can be a very distinguishable feature to identify a person. Having the color information in a separate channel (as is the case with the Hue Channel) makes it easier to extract the features and compare them. These features are tested in two ways: (1) the mean value of the Hue channel, both applied to the whole region of interest or to each of the 3 body parts; (2) a 9 histogram model, one for each channel and body part.

4.1.1.5 Texture Information

Texture information analyses the local differences in neighboring pixels and aggregates the changes in a measurable data model. Texture information can be complementary to color: an horizontal stripe pattern and a vertical stripe pattern with the same color are similar in color but very different in texture.

Texture has been used in appearance models along with color features in several algorithms, such as [40, 46].

4.1.1.6 CENTRIST

CENSus TRAnsform hISTogram (CENTRIST) is a visual descriptor that has been proposed for scene categorization [113]. The Census Transform (CT) is a local transform that was initially used to match image patches [114]. The Census Transform takes 3 by 3 patches and compares the intensities of the borders with the center, placing a bit 1 when the center pixel has a higher intensity when compared to the center or a 0 otherwise. Then, an 8 bit binary word is formed by grouping the 8 bits in a pre-determined sequence. An example is shown in Figure 4.2. The resulting values are then stored in a normalized histogram. Two tests are made: (1) a single histogram is used to model the whole region of interest; (2) the three body part model is tested.

$$\begin{array}{c|c|c} 32 & 64 & 96 \\ \hline 32 & \mathbf{64} & 96 \\ \hline 32 & 32 & 96 \end{array} \Rightarrow \begin{array}{ccc} 1 & 1 & 0 \\ 1 & 0 & \\ 1 & 1 & 0 \end{array} \Rightarrow (11010110)_2 \Rightarrow \mathbf{CT} = 214$$

Figure 4.2: Example of Application of CENTRIST to a 3x3 Window, extracted from [113]

4.1.1.7 Haar Wavelets

Haar Wavelets were originally used for people detection [115] but have recently been applied to people re-identification [116, 117] as texture descriptors. To apply the Haar Wavelets, the region of interest is transformed into four segments, which correspond to applying the different proposed Wavelets.

The resulting transformation is stored in a normalized histogram. Two tests are made: (1) a single histogram for the entire region of interest; (2) one histogram for each of the three body parts.

4.1.1.8 Edge Energy Descriptor

The Edge Energy Descriptor is presented in [118] and uses simple edge detectors based on Wavelets to find edges at 45° angles. The Edge Detectors are shown in Figure 4.3. For each of the pixels in the region of interest, the energy value is obtained by applying the block to the pixel and immediate neighbors. Each of the wavelets represents a different direction: vertical, horizontal and the two diagonals. The value of the pixels depends on how strong these edges are.

1	1	1	-1	$\sqrt{2}$	0	0	$\sqrt{2}$
-1	-1	1	-1	0	$-\sqrt{2}$	$-\sqrt{2}$	0

Figure 4.3: Edge Detectors for Edge Detection on 90° , 0° , 135° and 45° , extracted from [118]

To determine the energy value for the angle, the expression in Equation 4.6 is used. In the equation, θ represents the angle to test, M and N represent the dimensions of the region of interest and $e_{\theta(i,j)}$ represents the energy value for angle θ in pixel position (i, j) .

$$E_{\theta} = \sqrt{\frac{1}{MN} \sum_{i=0}^M \sum_{j=0}^N e_{\theta(i,j)}^2} \quad (4.6)$$

With this, a total of four values are obtained for each image: E_0 , E_{45} , E_{90} and E_{135} . To compare these values, the first step is to find the maximum energy value. The angle difference between the maximum energy values is used to define the values to compare: if on the first image the maximum is at the 45° angle and on the second image the maximum is at a 90° angle, then the 45° from the first image is compared to the 90° of the second; the 90° of the first is compared to the 135° of the second and so on. This step tries to ensure that the descriptor becomes rotation invariant. The distance result is the squared distance between the corresponding matches. A single test is made using the Edge Energy when applied to the entire region of interest. In this case, no 3 body part model is used because preliminary tests have shown no significant difference in the results.

4.1.1.9 Laplacian Operator

The Laplacian Operator has been widely used in the literature [119]. Equation 4.7 represents the operation made to each pixel in the image: to create the Laplacian frame, the second order derivative is used in each dimension.

$$Laplace(f) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (4.7)$$

Modifications on the Laplacian Operator have been proposed, including pyramid schemes [120]. These create multiple resolution versions of the image and apply the Laplacian Operator to each. In this case, however, since most regions of interest may have varying resolutions, the use of these pyramid models would be unreliable.

In practice, to apply the Laplacian Operator, the used formulation is the one from Equation 4.8: The center, top, bottom, left and right pixels are used to determine the new value. This operator is applied to each pixel region of interest.

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad (4.8)$$

After applying the Laplacian Operator, the information is stored in a histogram. Two tests are made: (1) single histogram; (2) three histograms, using the three body part model.

4.1.2 Local Features

Local Features, unlike the other methods presented, try to extract features from a limited set of regions. They are usually divided into two steps: detection and extraction. In the detection stage, relevant image keypoints are detected and on the extraction stage, the relevant features are extracted. These features can then be matched with points from other images and a comparison measure can be defined.

4.1.2.1 SIFT - Scale-Invariant Feature Transform

In the Scale-Invariant Feature Transform (SIFT) [35], extracted features are created as invariant to translation, scaling and rotation as well as partially invariant to any changes in illumination or geometric distortion.

Six different keypoints detectors are used: the SIFT detector, the FAST detector [121], a 1% grid, a 1% grid reduced to 70%, a 5% grid and a 5% grid reduced to 70% [122]. The test of several detectors tries to evaluate their use for people re-identification: the SIFT detector is computationally expensive; the FAST detector is faster, but may not extract enough points; the grid forces a series of points to be extracted, but they may not be relevant enough.

To compare the features, the distance function from [122] is used, which is presented as Equation 4.9. In it, d_i represents the distance between two points which are considered a match (found

by applying the Fast Approximate Nearest Neighbor Search [10]) and P_{max} is a constant that is higher than any of the individual distances. This constant is used to penalize unmatched points to avoid that completely different persons with a single perfect pair of points to be considered the same. N_1 and N_2 represent the number of extracted points from the two regions of interest. Finally, K represents the number of matches. This means that for each unmatched point a penalization of P_{max} is added. The distance function is then normalized to P_{max} .

$$D = \sum_{i=1}^K d_i + P_{max}[\max(N_1, N_2) - K] \quad (4.9)$$

4.1.2.2 SURF - Speeded-Up Robust Features

Speeded-Up Robust Features [32], or SURF, is a feature detector and feature extractor which aimed at providing a faster method when compared to other local features, hence the name Speeded-Up, while still being able to aggregate Robust information. The Distance function is again extracted from [122] and detailed in Subsection 4.1.2.1.

With SURF, 6 point detection methods are considered: the SURF detector, the FAST detector [121], a 1% grid, a 1% grid reduced to 70%, a 5% grid and a 5% grid reduced to 70% [122].

4.2 Testing Overview

The first tests were conducted with the objective of maximizing the 1st rank according to the Cumulative Matching Characteristic (CMC) [106] and on the CAVIAR4REID Dataset [27].

The CAVIAR4REID Dataset includes 10 to 20 images from each of the 72 persons included. The testing procedure for each of the features starts by selecting a random image for each person to create the model; Then, the remaining images are compared against all 72 created models. The re-identification algorithm then chooses the 10 closest models. If the best match is the right option, the 1st rank increases. If the second best match is the right option, the 2nd rank increases and so on. A simplified example with 3 persons is shown in Figure 4.4.



(a) Example of the 3 Images Used to Model the 3 Persons, 1 to 3



(b) Input Image Example 1



(c) Input Image Example 2

Figure 4.4: Re-Identification Example: Input images are compared to the available models in 4.4a and the best matches are ordered. For 4.4b, a possible vector could be [1 3 2], which means that the best match is Model 1, followed by 3 and 2; Since the image is from model 1, the 1st rank increases. For 4.4c, a possible vector could be [2 3 1]; Since the image is from model 3, the 2nd rank increases. The result would be a 1st rank of 50%, a 2nd rank of 50% and a 3rd rank of 0%. The CMC curve would have values 50%, 100% and 100%

To reduce noise from the background, the region of interest is limited by the interior ellipse of the region of interest, as shown in Figure 4.5.

The objective of this experiment is to identify which features show more promising results for further testing. A random decision would lead to a 1st rank of 1.39%. Each test was repeated 100 times to distribute the choice of the random models and average the comparison results. For each feature, the CMC curve for the first 10 ranks is presented.



Figure 4.5: Example Grayscale Images Using Ellipse Region of Interest

4.2.1 Global Information

4.2.1.1 Grayscale

The use of a mean intensity value resulted in a 1st rank of 1.7%, which is close to what a random decision would be. However, instead of using a single value, the 3 body part model can be used. Instead of using a distance between two values it uses all three values, as detailed in Equation 4.10.

$$Distance = w_1 \times Distance_{head} + w_2 \times Distance_{torso} + w_3 \times Distance_{legs} \quad (4.10)$$

With this, parameters w_1 , w_2 and w_3 need to be determined to maximize the results. This was done by varying each of the parameters from 0 to 1 in 0.01 increments, while keeping the sum at one (to maintain normalization). This test allows discarding one of the body parts (if they prove not to be relevant) and obtain the maximum value. The best combination of values is found in Equation 4.11, which achieves a 1st rank of 3.0% showing that the use of the 3 body part model has improved the re-identification rate. However, the low re-identification rate means that this feature may not be enough to deal with the changes in pose and noise that are common in these situations.

$$\begin{cases} w_1 = 17\% \\ w_2 = 48\% \\ w_3 = 35\% \end{cases} \quad (4.11)$$

Instead of extracting just the mean value, an alternative is extracting an histogram. Histograms may have a varying number of bins which group the information differently. To compare the histogram from an extracted model to the one from the input region of interest, the Chi-Square distance [123] is used. To find the optimal number of bins for the histogram, tests were made from 2 to 255 bins. The maximizing number of bins is 23, with a 1st rank result of 11.6%.

The results for a single histograms are already considerably higher than with the use of mean values (with an improvement of over 380%). Seeing the improvement of the use of the 3 part model, a similar test was conducted to determine the optimal weights, which can be seen in Equation 4.12. This results in a 1st rank of 15.5%. Surprisingly, the weight given to the head is considerably high since it's often considered to be the least identifiable body part [26] and in some cases completely discarded.

$$\begin{cases} w_1 = 24\% \\ w_2 = 38\% \\ w_3 = 38\% \end{cases} \quad (4.12)$$

The CMC Curve of these features up until the 10th rank is presented in Figure 4.6. The difference between the best result and the worst is over 10%, and over 15% of matches are made correctly on the first attempt. Also, within the first 10 ranks, over 40% of matches are correctly done using the best feature. It's also noticeable in the curvature on the first results for the histograms that it approximates the curve to an logarithmic curve, which means that it aggregates most results in the first matches.

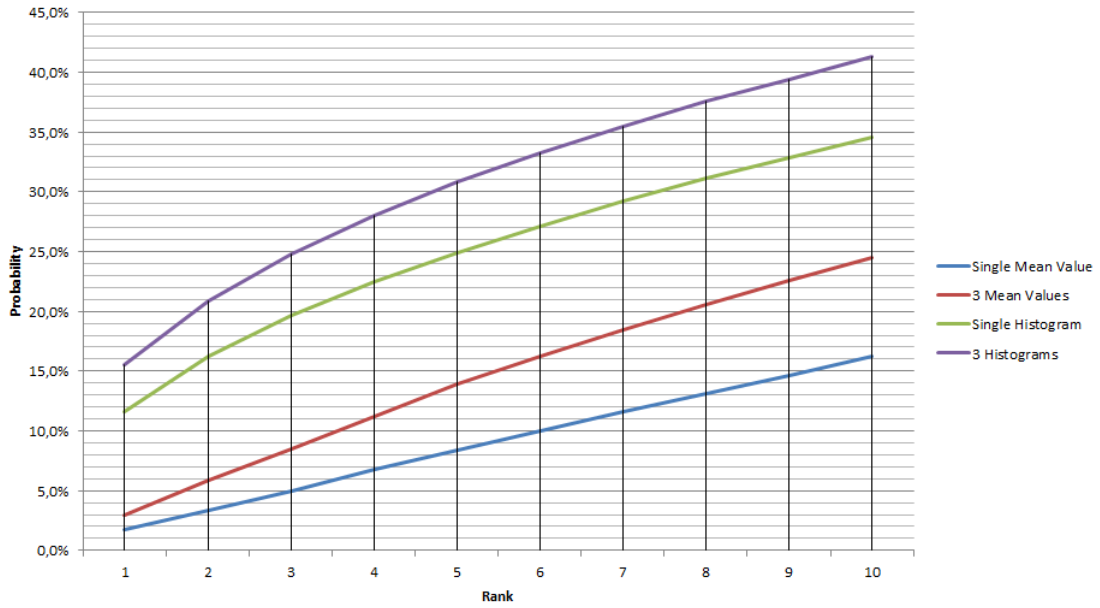


Figure 4.6: CMC Curve for Grayscale Features: The horizontal axis includes the different ranks; the vertical axis represents the probability of that particular rank. This representation will be maintained in future graphics

4.2.1.2 RGB

When using the RGB color space, 9 histograms are used. The number of bins is maintained at 23, which was determined in the grayscale case as the two colorspace are similar. The distance function is presented at Equation 4.13. The different parameters were tested to determine the best combination of the parameters which resulted in the set of w_1 , w_2 and w_3 seen in Equation 4.14. In this case, the head gets a very reduced weight, which is the normal consideration in literature. However, since the memory and computation cost is negligible in these operations, it is maintained. As for the channels, their weight distribution is shown in Equation 4.15. The red channel includes the largest weight, which means it can extract the more reliable features.

$$\begin{cases} Distance = c_1 \times Distance_{Red} + c_2 \times Distance_{Blue} + c_3 \times Distance_{Green} \\ Distance_{Red} = w_1 \times Distance_{Red,Head} + w_2 \times Distance_{Red,Torso} + w_3 \times Distance_{Red,Legs} \\ Distance_{Blue} = w_1 \times Distance_{Blue,Head} + w_2 \times Distance_{Blue,Torso} + w_3 \times Distance_{Blue,Legs} \\ Distance_{Green} = w_1 \times Distance_{Green,Head} + w_2 \times Distance_{Green,Torso} + w_3 \times Distance_{Green,Legs} \end{cases} \quad (4.13)$$

$$\begin{cases} w_1 = 7\% \\ w_2 = 58\% \\ w_3 = 35\% \end{cases} \quad (4.14)$$

$$\begin{cases} c_1 = 58\% \\ c_2 = 15\% \\ c_3 = 27\% \end{cases} \quad (4.15)$$

With this combination of values, the 1st rank is 20.6%. This means that a fifth of the results are correctly determined on the best match, an improvement on the grayscale features. The CMC curve for the RGB features is presented in Figure 4.7. It's also interesting to note that almost 50% of correct matches are made within the first 10 best attempts.

4.2.1.3 HSV

With the HSV colorspace, the Hue channel is of particular interest, as it is a representation of the color. Since color is what's being extracted, the mean hue value becomes a good candidate. When using a single hue value, a 1st rank re-identification rate of 2.8% is obtained. As it was the case with the grayscale value, the use of the 3 body part with the weights in Equation 4.16 has shown an improvement, with a 1st rank of 4.8%. As expected, since most relevant color information is in

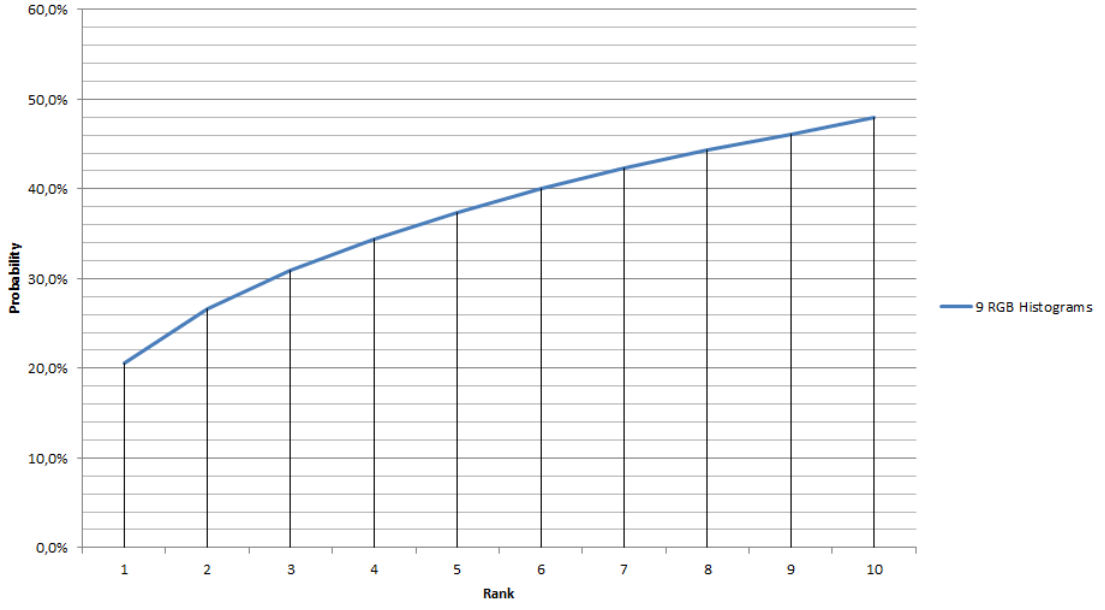


Figure 4.7: CMC Curve for 9 RGB Histograms

the torso and legs (since they make up for the clothing), the head weight is very low.

$$\begin{cases} w_1 = 4\% \\ w_2 = 49\% \\ w_3 = 47\% \end{cases} \quad (4.16)$$

Similarly with what was done in the RGB colorspace, a 9 HSV histogram test is made where the distance is defined as in Equation 4.17, with weights w_1 , w_2 and w_3 previously defined in Equation 4.16 and the channel weights being defined in Equation 4.18. The number of bins is again 23. Interestingly, the Hue Channel has a low value (less than a quarter), with the Value Channel, similar to what is found in the Grayscale colorspace, weighing over half.

$$\begin{cases} Distance = c_1 \times Distance_{Hue} + c_2 \times Distance_{Saturation} + c_3 \times Distance_{Value} \\ Distance_{Hue} = w_1 \times Distance_{Hue,Head} + w_2 \times Distance_{Hue,Torso} + w_3 \times Distance_{Hue,Legs} \\ Distance_{Saturation} = w_1 \times Distance_{Saturation,Head} + w_2 \times Distance_{Saturation,Torso} + w_3 \times Distance_{Saturation,Legs} \\ Distance_{Value} = w_1 \times Distance_{Value,Head} + w_2 \times Distance_{Value,Torso} + w_3 \times Distance_{Value,Legs} \end{cases} \quad (4.17)$$

$$\begin{cases} c_1 = 24\% \\ c_2 = 24\% \\ c_3 = 52\% \end{cases} \quad (4.18)$$

The 1st rank for the 9 HSV Histogram is 21.8%, which is very similar to the RGB result. The CMC curves for the HSV colorspace features is presented in Figure 4.8. With the 9 HSV histograms, over a fifth of matches are made correctly made on the first attempt.

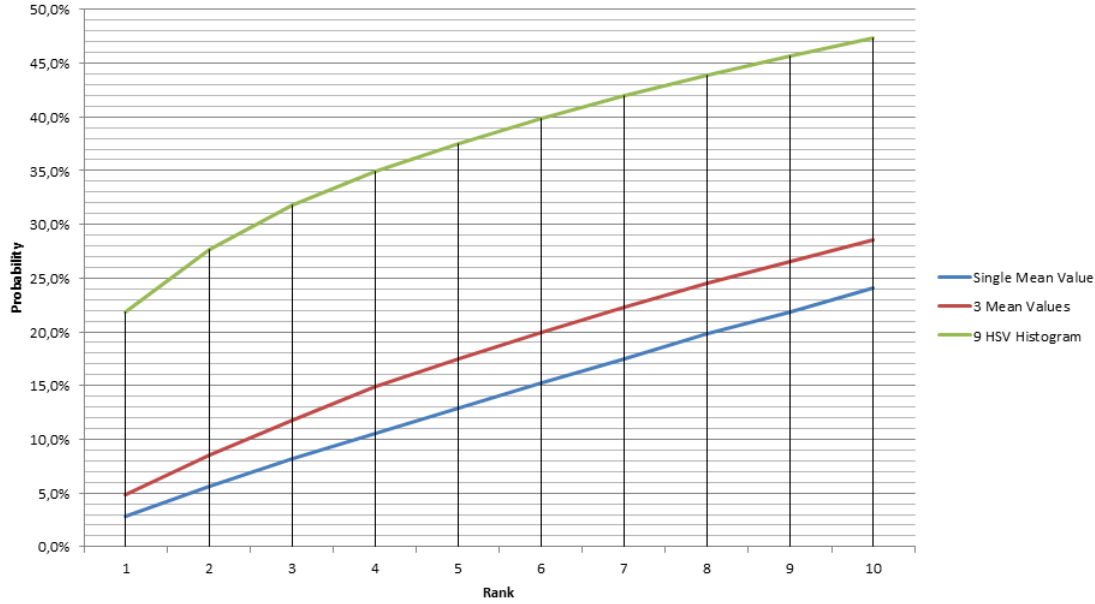


Figure 4.8: CMC Curve for HSV Features

4.2.1.4 CENTRIST

The first test using CENTRIST features was the optimization of the number of bins for the CENTRIST histogram. For this test, using a single histogram for each image, a varying number of bins was tested to maximize the 1st rank result. The optimal number of bins is 49, which provided a 1st rank result of 5.8%. While this result is lower than what's achieved with color histograms, it's not expected that texture features achieve better results but complement the color information. The 3 body part histogram follows the same structure of the color histograms and can be seen in Equation 4.10. The optimal weights for the body parts is seen in Equation 4.19, which achieves a 1st rank of 8.9%. The CMC curve for both CENTRIST tests is presented in Figure 4.9. There's again an improvement when using the 3 part body model with almost 9% of correct matches on the first attempt.

$$\begin{cases} w_1 = 7\% \\ w_2 = 58\% \\ w_3 = 35\% \end{cases} \quad (4.19)$$

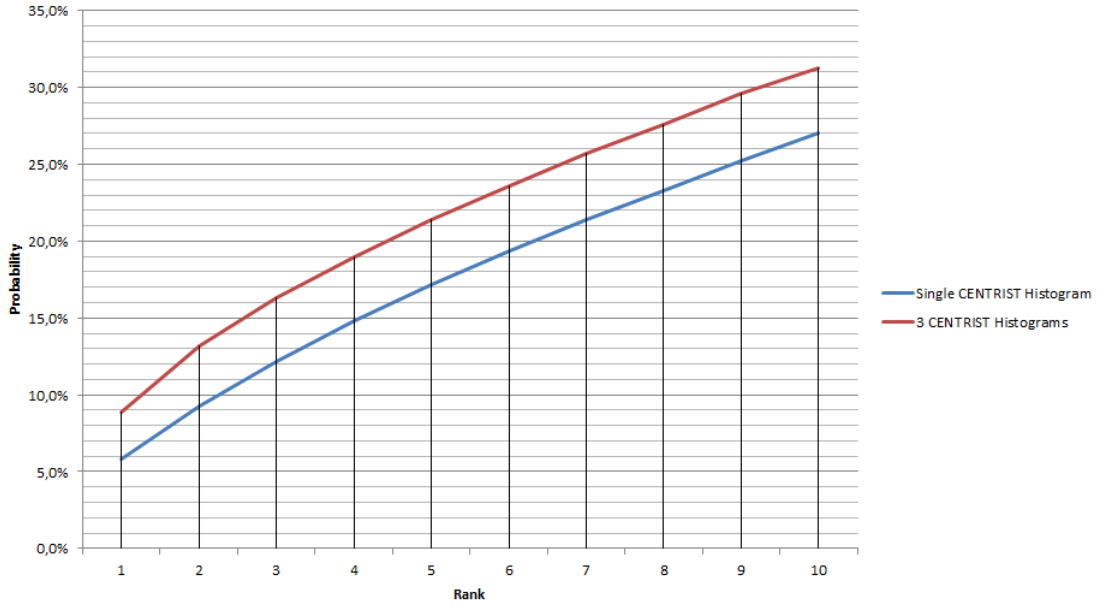


Figure 4.9: CMC Curve for CENTRIST Features

4.2.1.5 Wavelets

With the Wavelets, the maximizing number of bins was lower, at just 7, which means that this feature will take significantly less memory (which is not a relevant factor for these appearance models as histograms use a very small memory space). With a single histogram, the 1st rank result is 3.7%. When extending this model to the 3 body part using the weights in Equation 4.20. This shows a very even distribution of all weights, especially when compared to the CENTRIST, where the head would only amount to 7% of the decision. This leads to a 1st rank of 5.1%, almost half of what the 3 body CENTRIST achieved. The CMC Curve with the wavelet features is in Figure 4.10.

$$\begin{cases} w_1 = 22\% \\ w_2 = 37\% \\ w_3 = 41\% \end{cases} \quad (4.20)$$

4.2.1.6 Edge Energy Descriptor

The Edge Energy Descriptor is different from the other texture features as the information is not stored as a histogram, but as four values representing the edge intensity on four angles, with a more complex measure function. This configuration results in a 1st rank of 4.0%. With this feature, due to a 3 body part not being easily defined, no further testing is done. However, considering that the results are on par with the Wavelets, it would not be expected for the results to suffer a big improvement. The CMC Curve is presented in Figure 4.11, which even on the 10th rank it's still barely over a fifth of correct chances.

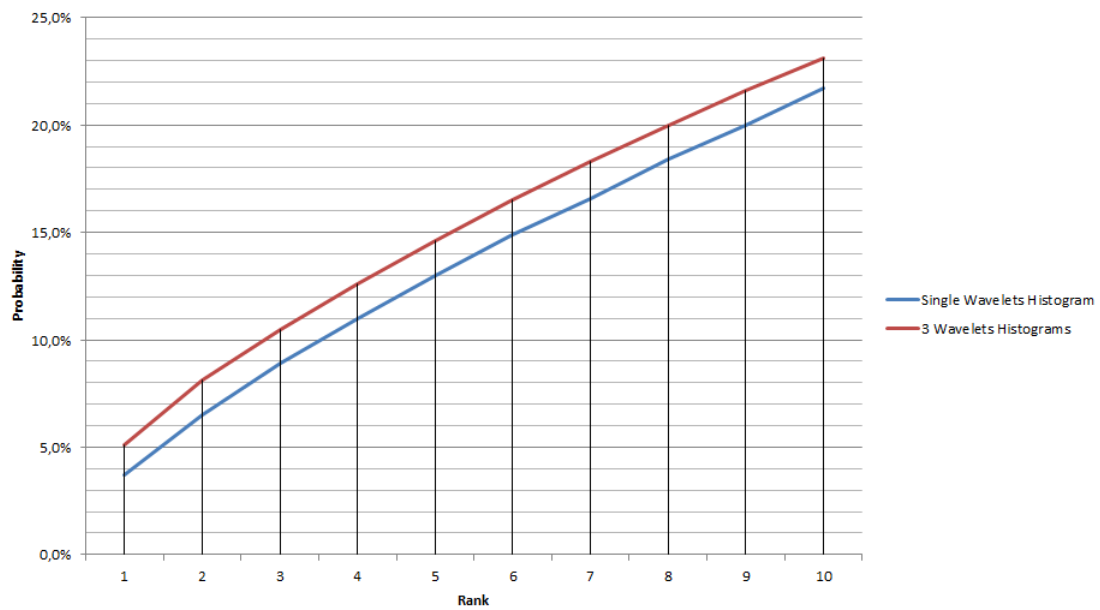


Figure 4.10: CMC Curve for Wavelet Features

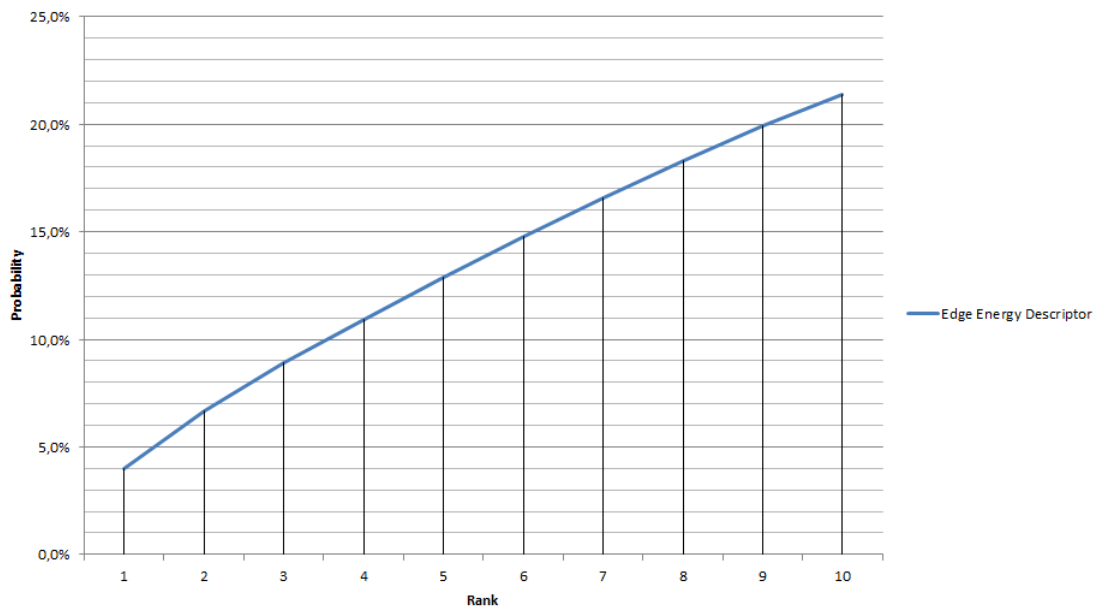


Figure 4.11: CMC Curve for Edge Energy Feature

4.2.1.7 Laplacian Operator

For the use of the Laplacian Operator, the histogram representation is again used. First, the optimal number of bins was determined to be 24, which resulted in a 1st rank of 5.4%, which is close to the use of the single CENTRIST histogram. A similar extension of the 3 body model was made, with the weights from Equation 4.21. Interestingly, the weight distribution is similar with CENTRIST, which likely means that most texture information is in the torso and almost no texture information is in the head. This extension lead to a 1st rank of 6.7%. Comparing to the improvement in the CENTRIST, which reached 8.9%, the result is lower but still the second best texture 1st rank. The CMC curve for Laplacian features is presented in Figure 4.12. The improvement of using the 3 body part model is more noticeable on higher ranks, which means that this model is better at aggregating the results on the first matches.

$$\begin{cases} w_1 = 7\% \\ w_2 = 55\% \\ w_3 = 38\% \end{cases} \quad (4.21)$$

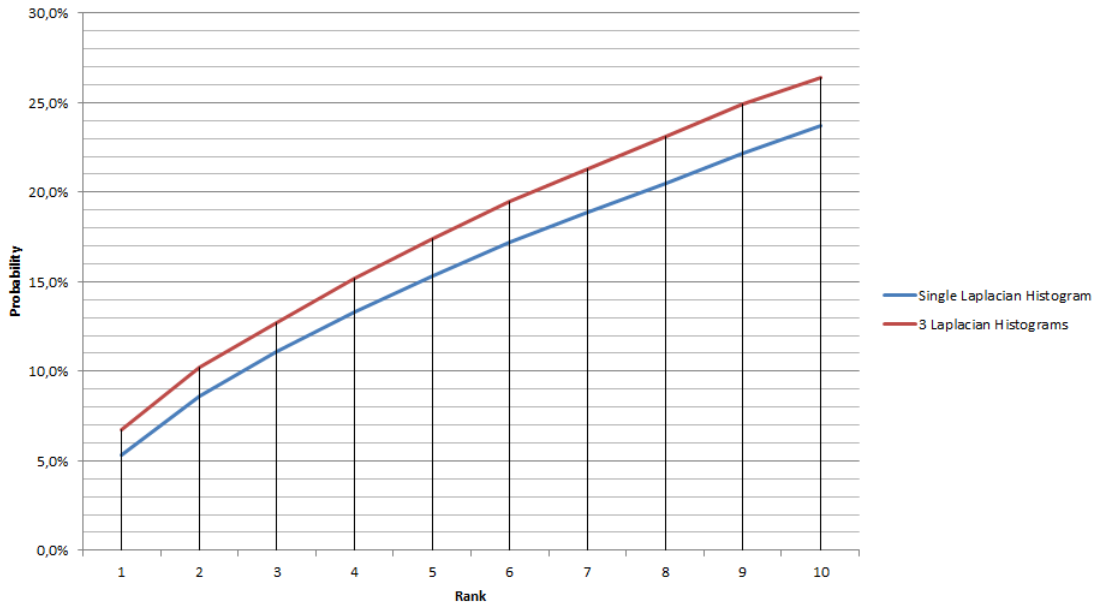


Figure 4.12: CMC Curve for Laplacian Features

4.2.2 Local Information

4.2.2.1 SIFT

For SIFT (and later SURF) features, instead of representing the results on a CMC graphic, a table representation is used because there are several close results which would make a graphical representation hard to read. Table 4.1 represents the rank results for the tested detectors, where (*R*) represents the result for a reduced grid. The FAST detector results are the best in all ranks, with a tenth of the matches being correctly made on the first try, which is better than any of the tested texture features but smaller than the best results for each of the color features.

Rank	SIFT Detector	FAST Detector	1% Grid	1% Grid (<i>R</i>)	5% Grid	5% Grid (<i>R</i>)
1	6.6%	10.5%	1.8%	2.3%	1.9%	2.2%
2	8.8%	12.6%	3.5%	4.5%	3.6%	4.0%
3	10.6%	14.2%	5.5%	6.8%	5.2%	6.0%
4	12.4%	15.8%	7.8%	8.6%	7.0%	7.8%
5	14.0%	17.2%	9.6%	10.6%	8.6%	9.5%
6	15.5%	18.5%	10.9%	12.1%	10.2%	11.2%
7	16.8%	19.9%	12.3%	13.6%	11.1%	12.5%
8	18.3%	21.4%	14.2%	15.6%	12.7%	14.3%
9	19.8%	22.7%	15.8%	17.8%	14.4%	15.8%
10	21.4%	23.9%	17.6%	19.2%	15.8%	17.2%

Table 4.1: Rank Values for SIFT Features

4.2.2.2 SURF

The rank results for SURF extracted features are much closer with each other than with SIFT, with one detector (SURF) being the best for 1st rank at 5.1% but the 2nd and 3rd ranks being higher on the 1% Grid and the remaining ranks on the reduced 5% Grid. The full results are presented in Table 4.2. Once again (*R*) represents the use of a reduced grid of points. The results with the SURF extractor are about half of what SIFT features reach.

Rank	SURF Detector	FAST Detector	1% Grid	1% Grid (<i>R</i>)	5% Grid	5% Grid (<i>R</i>)
1	5.1%	4.5%	4.9%	4.7%	4.2%	4.5%
2	7.5%	7.4%	8.2%	7.6%	7.0%	7.7%
3	9.6%	9.9%	10.5%	10.0%	9.4%	10.2%
4	11.4%	12.2%	12.5%	12.3%	11.6%	12.8%
5	13.0%	14.5%	15.0%	14.1%	13.6%	15.5%
6	14.8%	16.7%	17.1%	15.9%	15.6%	17.7%
7	16.4%	18.6%	18.9%	18.0%	17.4%	19.8%
8	18.0%	20.4%	20.6%	19.6%	19.2%	21.7%
9	19.6%	22.3%	22.2%	21.4%	20.9%	23.4%
10	21.2%	24.0%	23.8%	23.2%	22.2%	25.2%

Table 4.2: Rank Values for SURF Features

4.2.3 Result Overview

Up until this point, 27 different tests were made, with 8 for color features, 7 for texture features and 12 for local features. Tables 4.3, 4.4 and 4.5 represent the 1st rank for each of the features.

Color Features	First Rank (%)
Mean Grayscale Intensity Value	1.7%
3 Mean Grayscale Intensity Values	3.0%
Grayscale Histogram	11.6%
3 Grayscale Histograms	15.5%
9 RGB Histograms	20.6%
Mean Hue Value	2.8%
3 Mean Hue Values	4.8%
9 HSV Histograms	21.8%

Table 4.3: 1st Rank Results for Tested Color Features

Texture Features	First Rank (%)
1 CENTRIST Histogram	5.8%
3 CENTRIST Histograms	8.9%
1 Wavelets Histogram	3.7%
3 Wavelets Histograms	5.1%
4 Edge Energy Values	4.0%
1 Laplacian Histogram	5.3%
3 Laplacian Histograms	6.7%

Table 4.4: 1st Rank Results for Tested Texture Features

Color features represent the best overall results, with 3 features surpassing the 15% mark. While the 9 HSV Histograms and the 3 Grayscale Histograms share some redundant information, they are both analyzed to verify if they work well for different types of persons to re-identify.

The results on texture features aren't as high as the results in color features. This is expected: not all images have enough texture to be easily recognizable, and the presence of noise or low resolutions reduce the amount of texture information that can be extracted. Two of the highest ranking features are selected for analysis: 3 CENTRIST histograms and 3 Laplacian Histograms. This corresponds to choosing the two best texture features found.

Because of the low results of the SURF features, with no result being over 5%, only the best SIFT feature is analyzed. Even though there's another SIFT extracted feature that's similar with the 3 Laplacian Histograms, having two SIFT features would be redundant, since it would extract the same type of information.

Local Features	First Rank (%)
SURF with SURF Detector	5.1%
SURF with FAST Detector	4.5%
SURF with 1% Grid	4.9%
SURF with 1% Grid on 70% Reduced Image	4.7%
SURF with 5% Grid	4.2%
SURF with 5% Grid on 70% Reduced Image	4.5%
SIFT with SIFT Detector	6.6%
SIFT with FAST Detector	10.5%
SIFT with 1% Grid	1.8%
SIFT with 1% Grid on 70% Reduced Image	2.3%
SIFT with 5% Grid	1.9%
SIFT with 5% Grid on 70% Reduced Image	2.2%

Table 4.5: 1st Rank Results for Tested Local Features

A total of 6 features are analyzed: 3 color features, 2 texture features and 1 local feature, which have the 1st rank seen in Table 4.6.

Feature	First Rank (%)
3 Grayscale Histograms	15.5%
9 RGB Histograms	20.6%
9 HSV Histograms	21.8%
3 CENTRIST Histograms	8.9%
3 Laplacian Histograms	6.7%
SIFT with FAST Detector	10.5%

Table 4.6: 1st Rank Results for Analyzed Features

4.3 Detailed Analysis

The main objective of this section is to analyze how the selected features work for each of the 72 individuals in the dataset and to filter out features that only work on a very small subset of individuals. It's also important to verify the person distribution: if all features have a uniform distribution of results, combining them wouldn't improve the results (or would have a very reduced effect). If, on the other hand, this distribution isn't uniform and the distributions don't match their maximum values, features can work together on the re-identification.

For this section, two graphics are presented for each feature: the first one indicates the 1st rank result for that feature for each person. It'll help determine the distribution of probabilities and if some methods only work on persons with a very specific set of characteristics. The second graphic is a confusion matrix and it gives the probability of the person X (line) to be identified as the person Y (column). Ideally, only the diagonal of the matrix is seen. The analysis of this graphic can be used to find persons who are often confused with others. To make it easier to read, no probability values are shown, but instead a color scheme where white represents 0% and black represents 100% is used.

4.3.1 Global Information

4.3.1.1 Grayscale

The 1st rank result for each of the persons is presented in Figure 4.14, which shows that there are very few persons where the probability is under 10% (4 persons), which is a good indicator of the reliability of the feature as no person is expected to be re-identified less than 10% of times. It also shows some cases where the re-identification rate is over 50% (4 persons). There are still some persons with a relatively low re-identification rate (7.3%). The confusion matrix is shown in Figure 4.13, in which the diagonal is very clear but also includes several other dark spots. One noticeable thing is that when a the diagonal has a low probability, the column usually doesn't have any high value. This means that while the person is not being correctly identified, it's not being confused with any other person in particular.

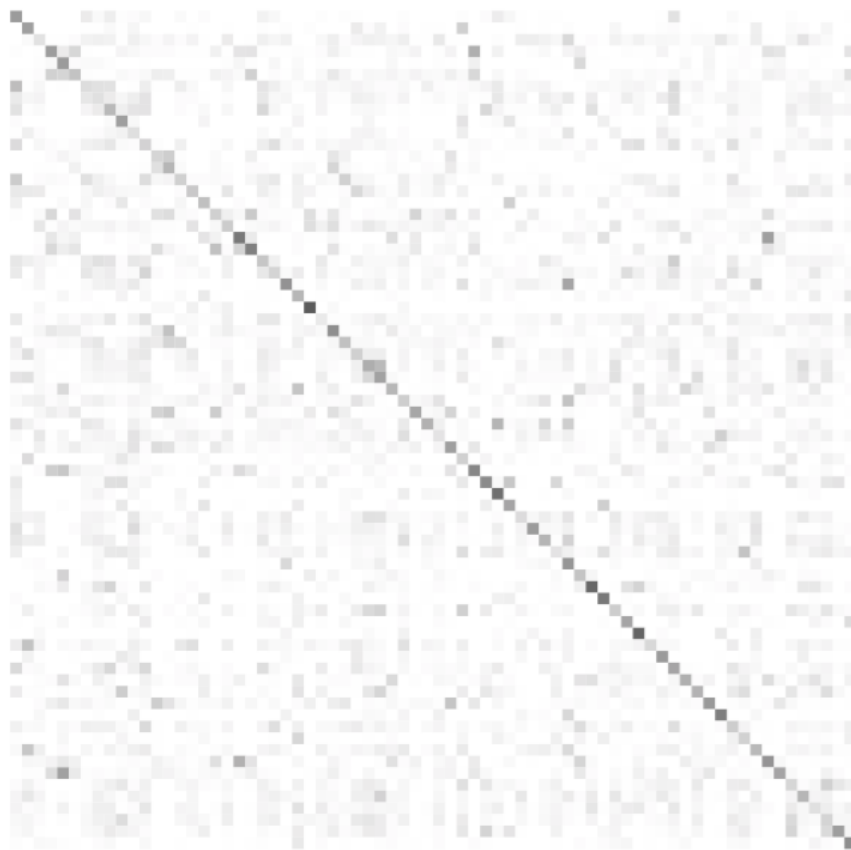


Figure 4.13: Visual Confusion Matrix Representation for Grayscale Features. Lighter colors represent low probabilities; Darker colors represent high probabilities

Three examples of the person with the best re-identification are shown in Figure 4.15. This example shows the type of individuals that gets good results: a clear grayscale color and little variation. It's also likely that the occlusion in some of the available images for this person decreases the results. Three examples of the person with the worst re-identification are shown in Figure

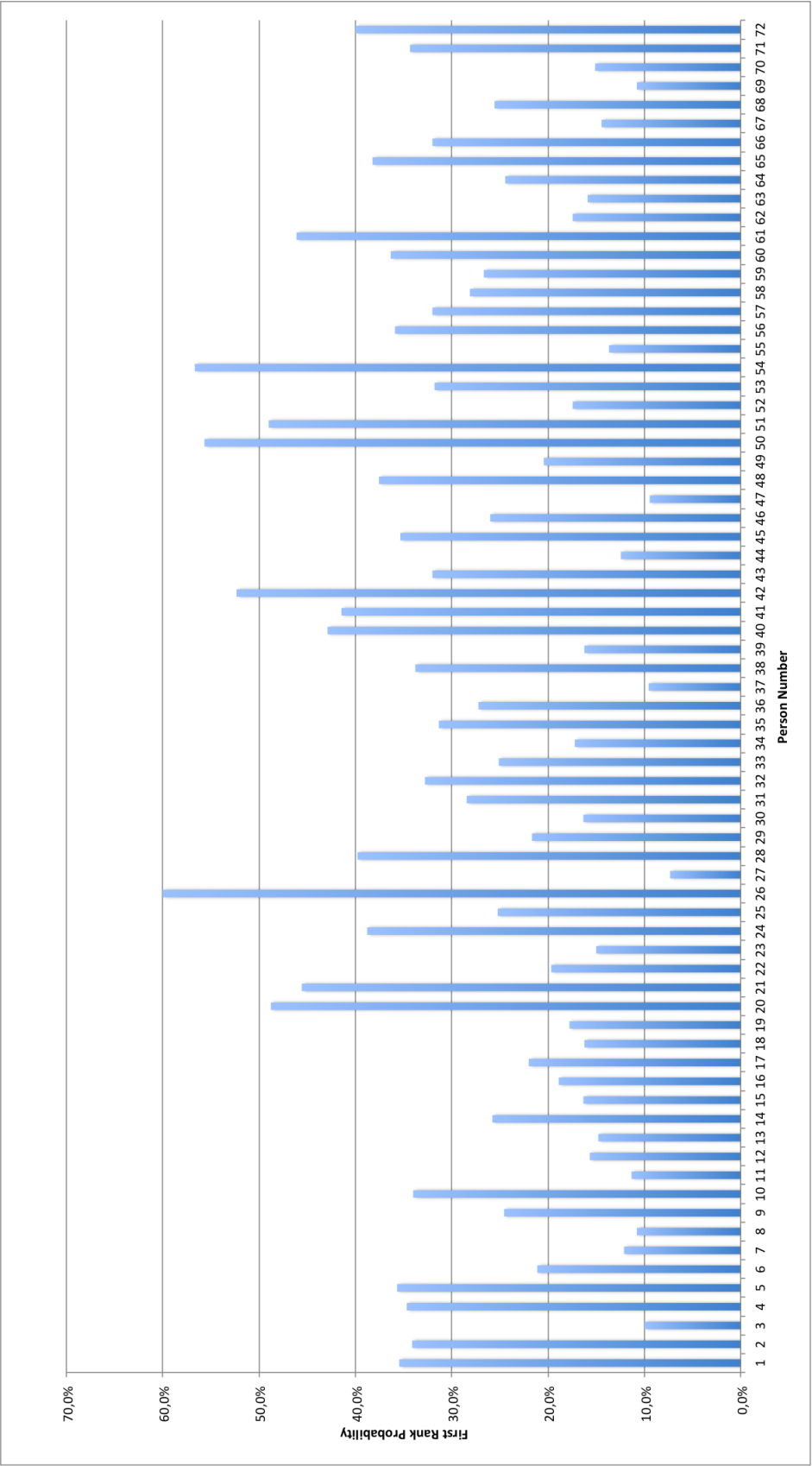


Figure 4.14: 1st Rank Result for Each Person Using 3 Body Part Grayscale Histograms

4.16. In this case, since the perspectives are quite different, the corresponding model would be very different. Additionally, there are several other persons with a similar grayscale configuration.



Figure 4.15: Best Person for Grayscale Re-Identification



Figure 4.16: Worst Person for Grayscale Re-Identification

4.3.1.2 RGB

For the 3 body part RGB histogram, the 1st rank result for each of the persons is presented in Figure 4.18. Comparing to the results in the grayscale case, there are more people which have a re-identification rate of under 10% (9 persons) but also more people who are correctly re-identified in over 40% of occasions (10 persons). This shows that the RGB features aren't as consistent and reliable as grayscale features. The visual confusion matrix is presented in Figure 4.17. In this case, the diagonal is more varied, with very light and very dark colors, which again shows the inconsistency of the feature. As it was the case with grayscale features, no single person is particularly confused with another, with an even distribution of errors.

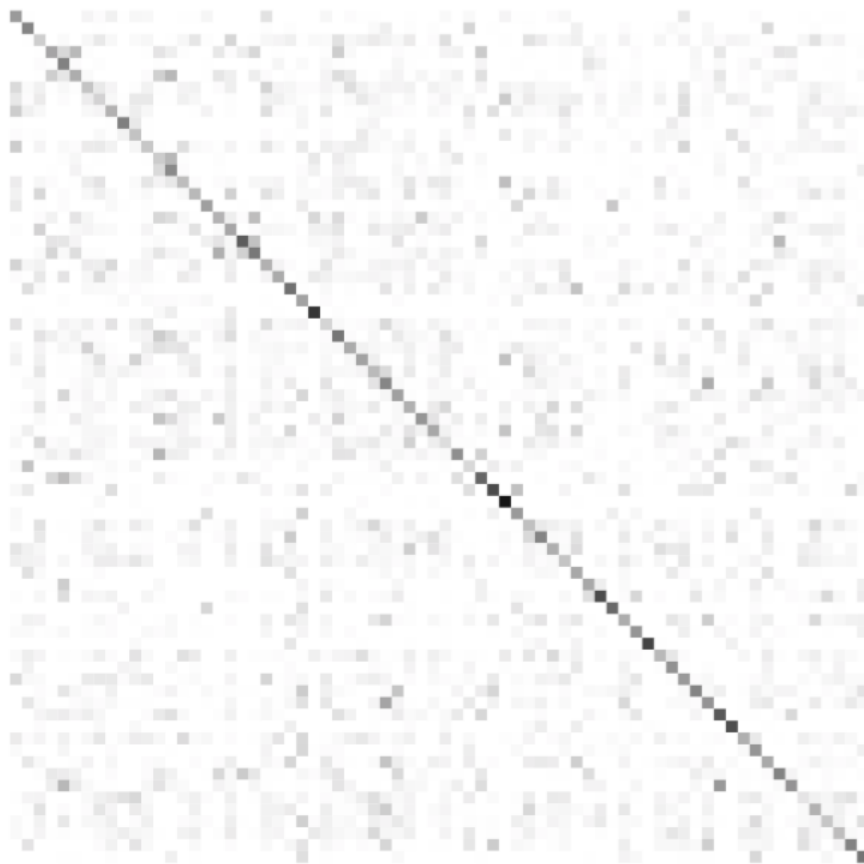


Figure 4.17: Visual Confusion Matrix Representation for RGB Features. Lighter colors represent low probabilities; Darker colors represent high probabilities

Three examples of the person with the best re-identification rate are shown in Figure 4.19. This person is always presented in a similar pose and with a very clear color structure, which explains the 91.8% re-identification rate. On the other hand, three examples of the person with the worst re-identification rate are shown in Figure 4.20. The changes in resolution, pose and the dark color make re-identification using RGB histograms much harder.

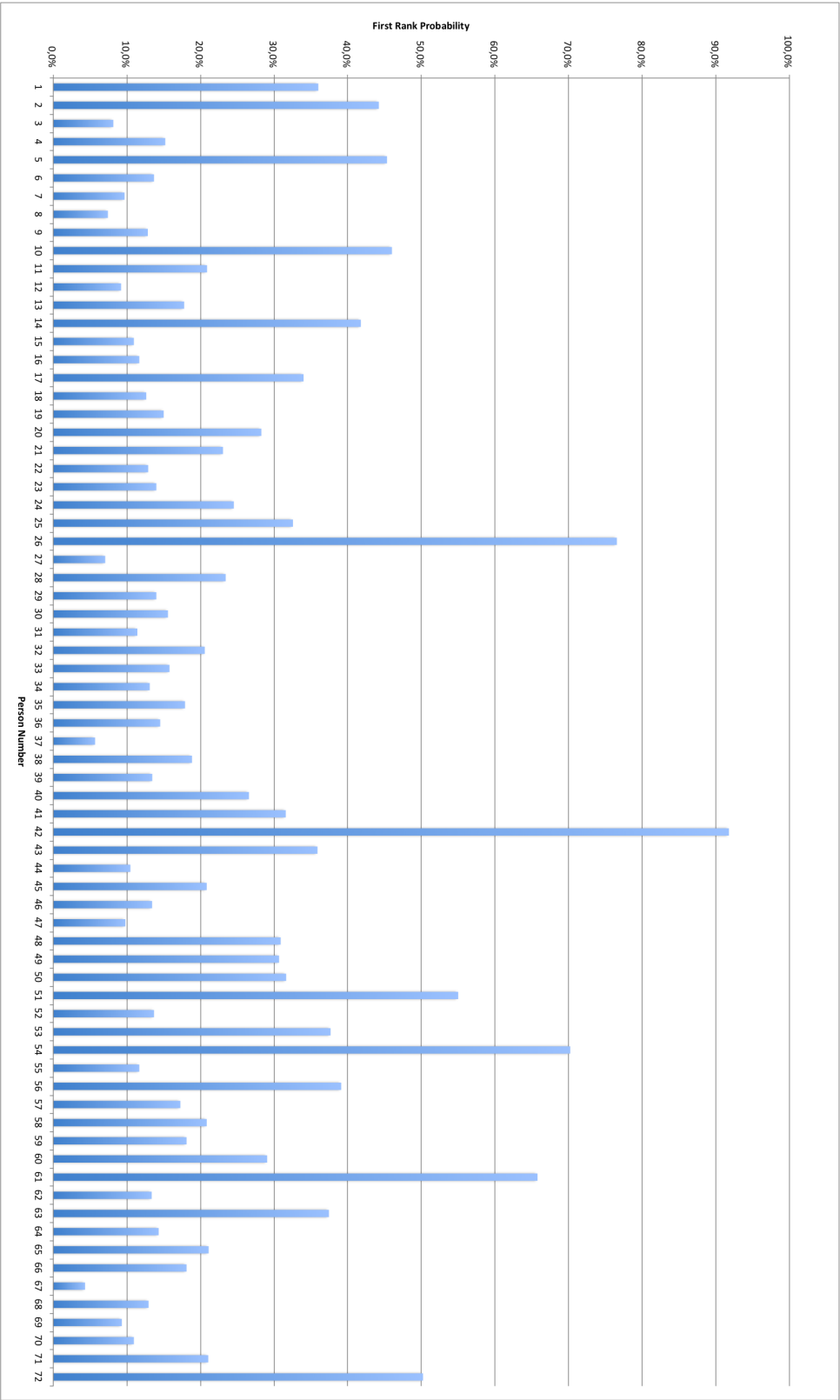


Figure 4.18: 1st Rank Result for Each Person Using 3 Body Part RGB Histogram

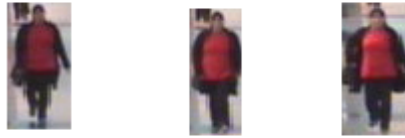


Figure 4.19: Best Person for RGB Re-Identification



Figure 4.20: Worst Person for RGB Re-Identification

4.3.1.3 HSV

For the 3 body part HSV histogram, the 1st rank result for each of the persons is presented in Figure 4.23. When compared to the RGB histograms, this is a more even distribution in the sense that there are less low results (only 5 persons under the 10% re-identification mark) even though it has several peaks (11 persons over 40%). This means that the HSV histogram can deal with a wider range of different persons. The visual confusion matrix is presented in Figure 4.21 and shows a very clear diagonal. Persons 8 and 15 are often re-identified as person 43. This case, seen in Figure 4.22, the persons all have a red colored torso and dark legs, with an overall similar color feature.

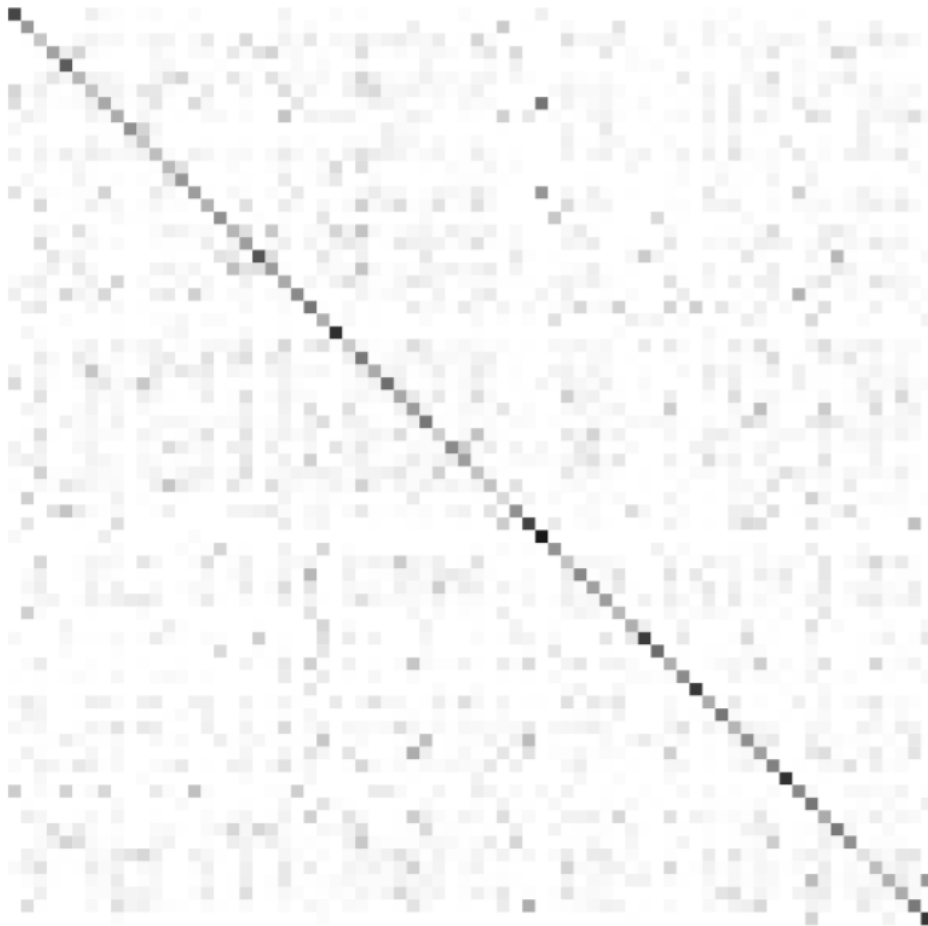


Figure 4.21: Visual Confusion Matrix Representation for HSV Features. Lighter colors represent low probabilities; Darker colors represent high probabilities

The highest result is high (90.2%) but there are very low results (4.7%). The person with the highest and lowest probabilities are the same as in RGB histograms, which is expected since they are both color features.

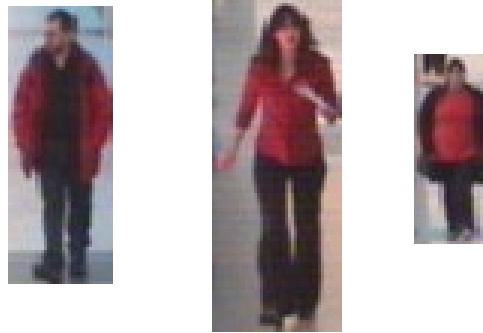


Figure 4.22: Example of persons confused with HSV features

4.3.1.4 Color Analysis Overview

After analyzing the different color features, grayscale and HSV are the most reliable of the two. HSV especially proves to get very good results since it has very few persons with a re-identification rate under 10%. One thing of interest is that even though some of the best and worst matches are the similar on several of these color features, there are a few differences which means that using them together may get better results than using them individually.

While the RGB features were inconsistent in some persons, they will be added to the proposed appearance model.



Figure 4.23: 1st Rank Result for Each Person Using 3 Body Part HSV Histogram

4.3.1.5 CENTRIST

For the 3 body part CENTRIST histogram, the 1st rank result for each of the persons is presented in Figure 4.25. When comparing to any of the color features, the results are lower (which is expected, since the average is also lower), but the distribution shows that it's very uniform. However, in most persons, the re-identification rate is under 5% and the persons with higher re-identification often match the higher re-identifications of color features. The visual confusion matrix is presented in Figure 4.24, which shows a lighter diagonal line but also some vertical lines, which indicate several cases where some persons are often poorly re-identified.

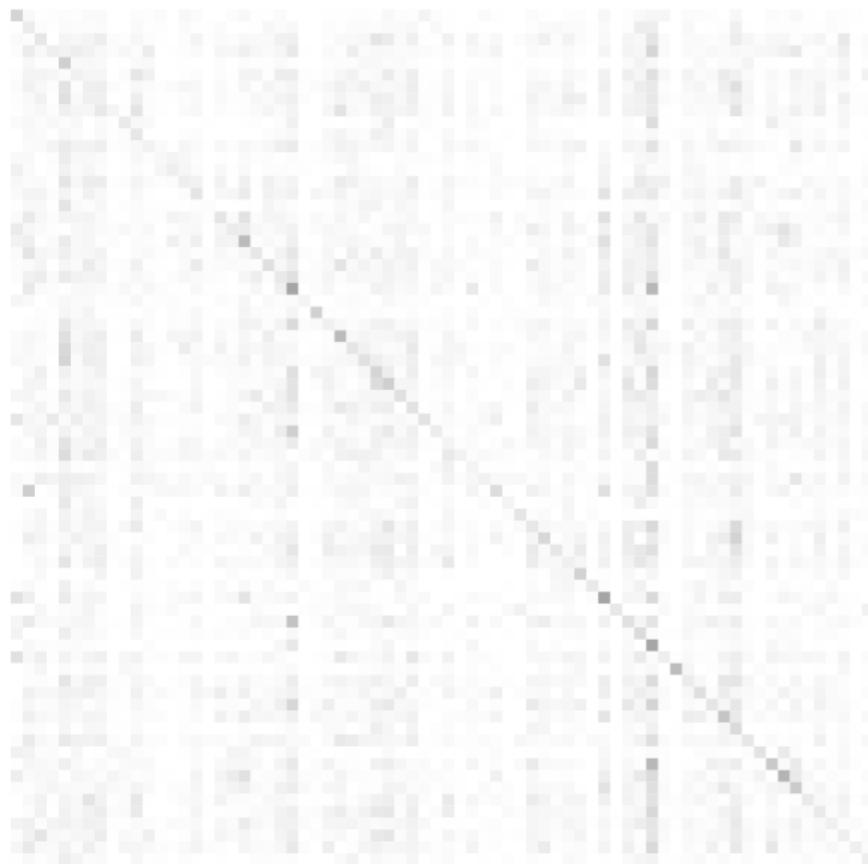


Figure 4.24: Visual Confusion Matrix Representation for CENTRIST Features. Lighter colors represent low probabilities; Darker colors represent high probabilities

The 1st rank result of the best person is 35.0% and the lowest is 1.2%. Three examples of the person with the best re-identification rate are shown in Figure 4.26. Even though there are significant changes in pose, the CENTRIST histograms prove to be quite pose invariant. Three examples of the worse cases are shown in Figure 4.27. for this person, only low resolution images are available, which make extracting texture features impossible due to the lack of detail.

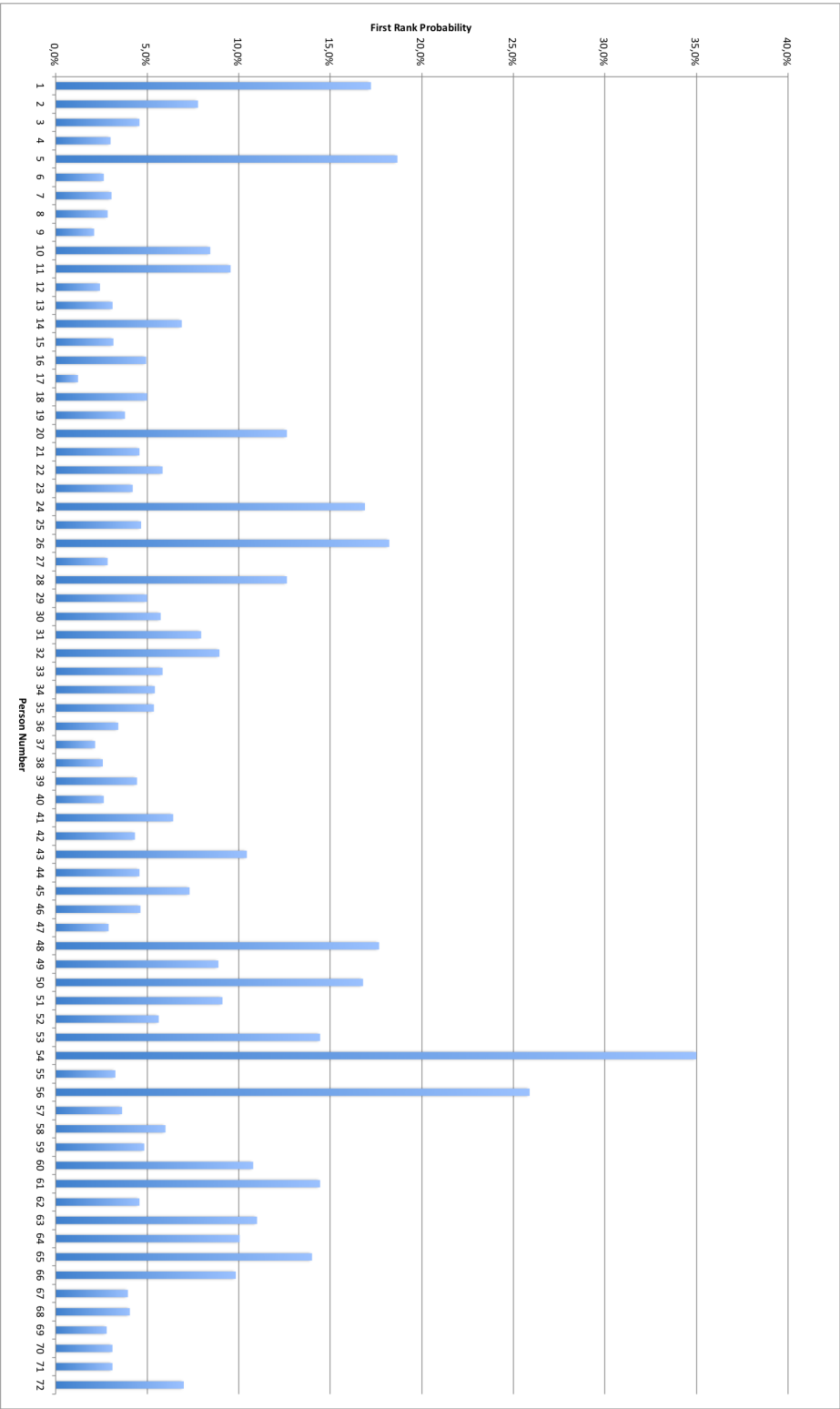


Figure 4.25: 1st Rank Result for Each Person Using 3 Body Part CENTRIST Histogram



Figure 4.26: Best Person for CENTRIST Re-Identification



Figure 4.27: Worst Persons for CENTRIST Re-Identification

4.3.1.6 Laplacian

For the 3 body part Laplacian histogram, the 1st rank result for each of the persons is presented in Figure 4.29. In this case again most results are under 5% but compared to the CENTRIST histogram, these are some cases where the CENTRIST did worse than the Laplacian Histograms gets a better result, such as person 72. It is expected that using them together will improve the results when compared to using a single feature. Figure 4.28 shows that there are very low, distributed values, with the diagonal not being very visible.



Figure 4.28: Visual Confusion Matrix Representation for Laplacian Features. Lighter colors represent low probabilities; Darker colors represent high probabilities

As it was the case with CENTRIST, some persons have a good re-identification rate while others have an extremely low 1st rank. The best result, of 46.9%, is achieved in the same person as the CENTRIST, for similar reasons. Three examples of the worst case are shown in Figure 4.30. The low resolution of most images of the person leads to the poor result of 1.2%.

4.3.1.7 Texture Analysis Discussion

From the deeper analysis on the texture, it's worth noting that the persons in which one texture descriptors works best and worst aren't the same. This means that using both together can be used

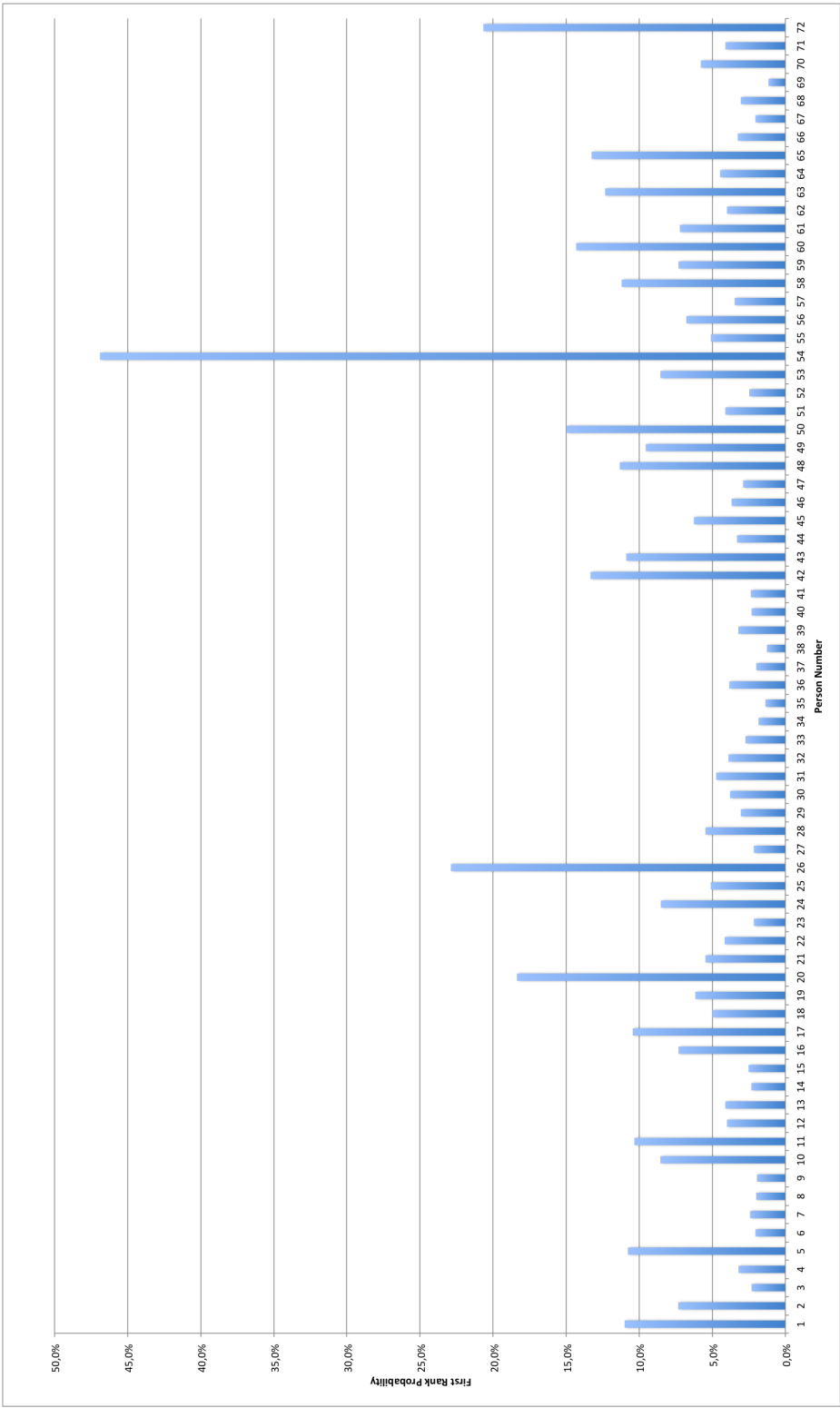


Figure 4.29: 1st Rank Result for Each Person Using 3 Body Part Laplacian Histogram



Figure 4.30: Worst Person for Laplacian Re-Identification

to improve the overall results. Still, the texture results are overall worse than the color features. This was expected mainly for two reasons: (1) the low resolution of the images means that there is not enough room for details, which are important to extract texture features; (2) the noise from the pictures also reduced the amount of texture information that can be extracted.

Still, both CENTRIST and Laplacian features will be part of the test on the appearance model.

4.3.2 Local Information

4.3.2.1 SIFT

For SIFT Features with the FAST Detector, the 1st rank result for each of the persons is presented in Figure 4.32. These features show high re-identification rates on the same persons as the previous global methods. While the overall results aren't worst than the texture features, they may not add any relevant information. The main issue is seen in the confusion matrix of Figure 4.31, which shows that most persons are being re-identified as person 1 or 2.

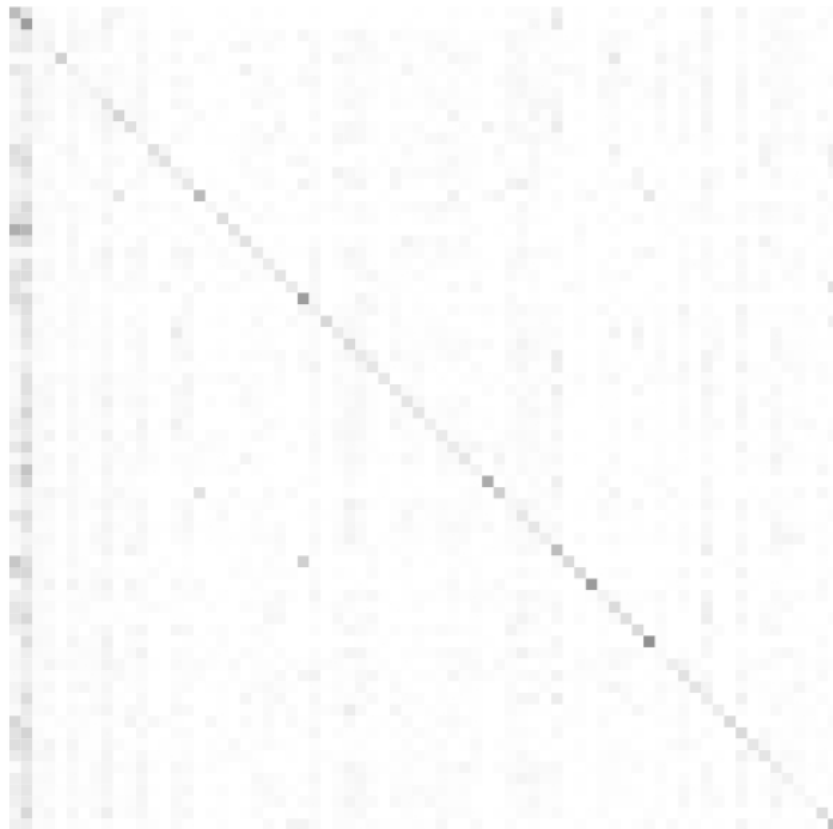


Figure 4.31: Visual Confusion Matrix Representation for SIFT Features. Lighter colors represent low probabilities; Darker colors represent high probabilities

4.3.2.2 Local Features Analysis Overview

The combination of SIFT features with the FAST Detector has shown good overall results similar to those found in the texture features. The main problem with it is that it seems to add too much noise in the models, with the incorrect re-identifications of most persons as persons 1 and 2. Additionally, local features take longer to detect and extract.

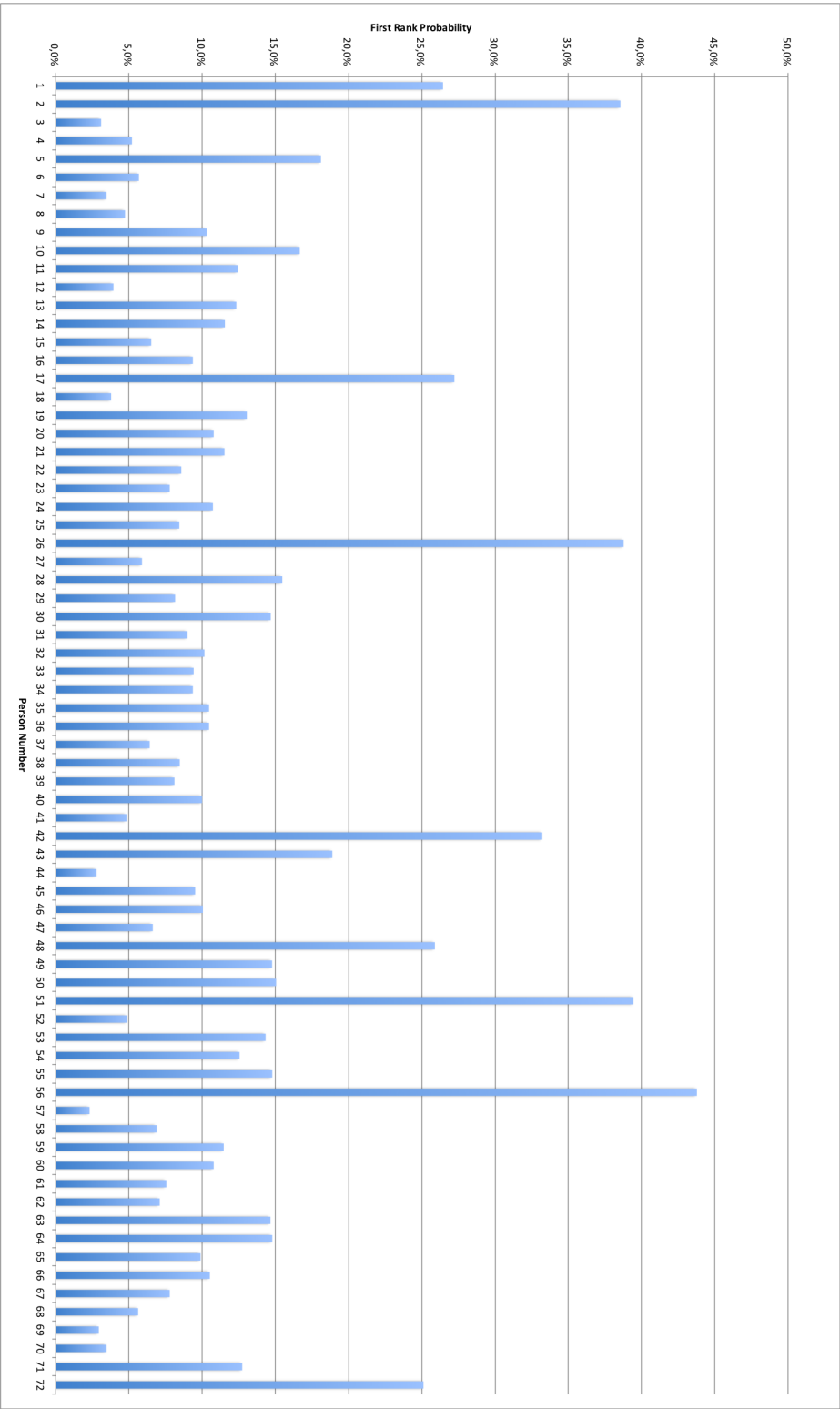


Figure 4.32: 1st Rank Result for Each Person Using SIFT Features with the FAST Detector

4.4 Conclusions

Placing the different tested methods side by side for each of the persons, in Figures 4.33 and 4.34, it's clear that the results of each method are different for each person. Additionally, there's a tendency of similar methods (Grayscale/RGB/HSV or CENTRIST/Laplacian) to have more similar results. Seeing the methods side by side, if the system could correctly choose which type of feature to use, the results could be greatly improved (case where the feature with the highest re-identification result is chosen for each person). Comparing the results, there is no single method better than all the other, and a combination of methods is the best course of action. There are cases, such as person 37, that none of the tested features reach 10%, in which re-identification is not expected to significantly increase. In general, color features work better than texture features and will probably have a bigger weight in the final model.

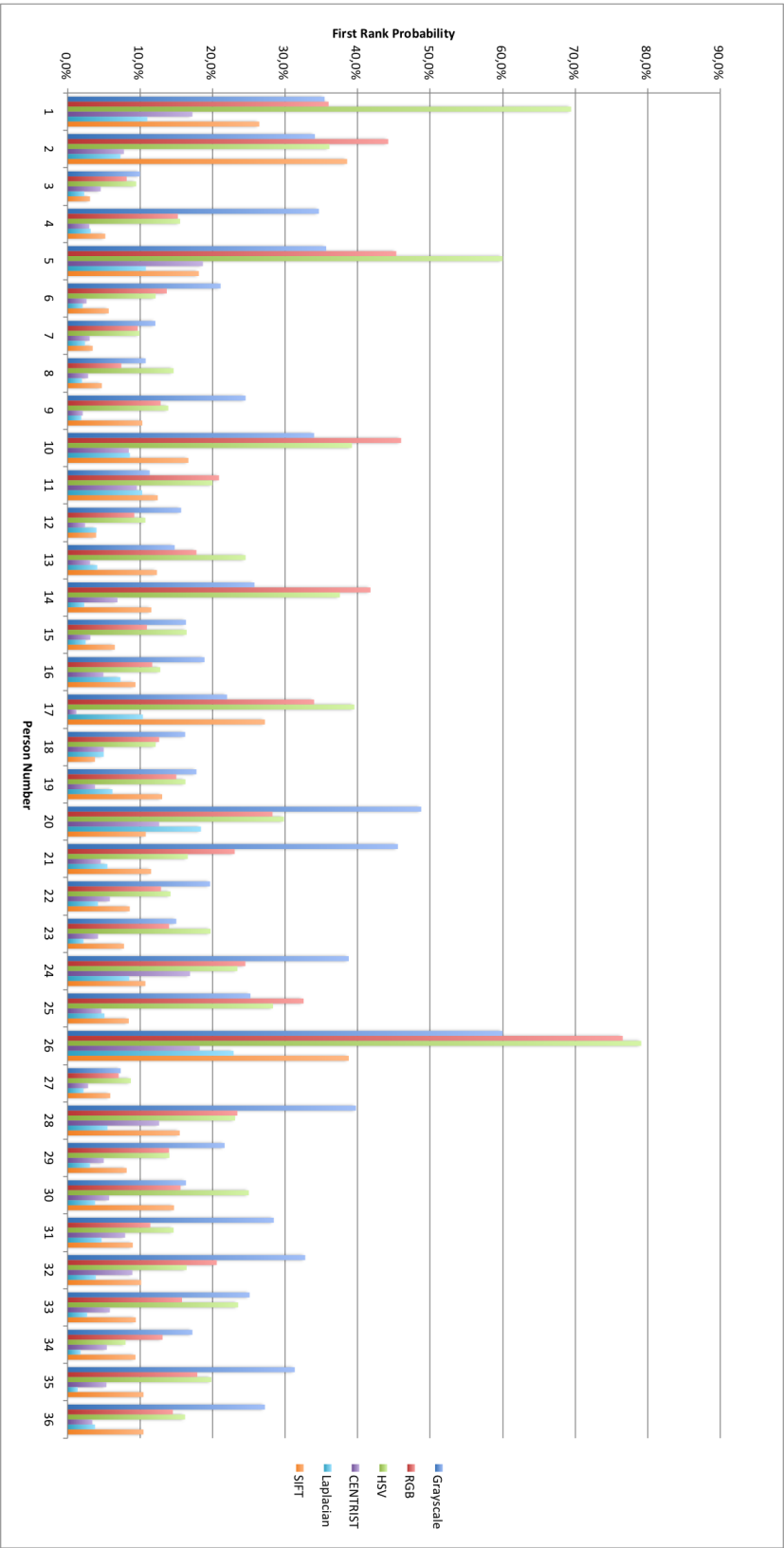


Figure 4.33: 1st Rank Result for Each Person and Each Method

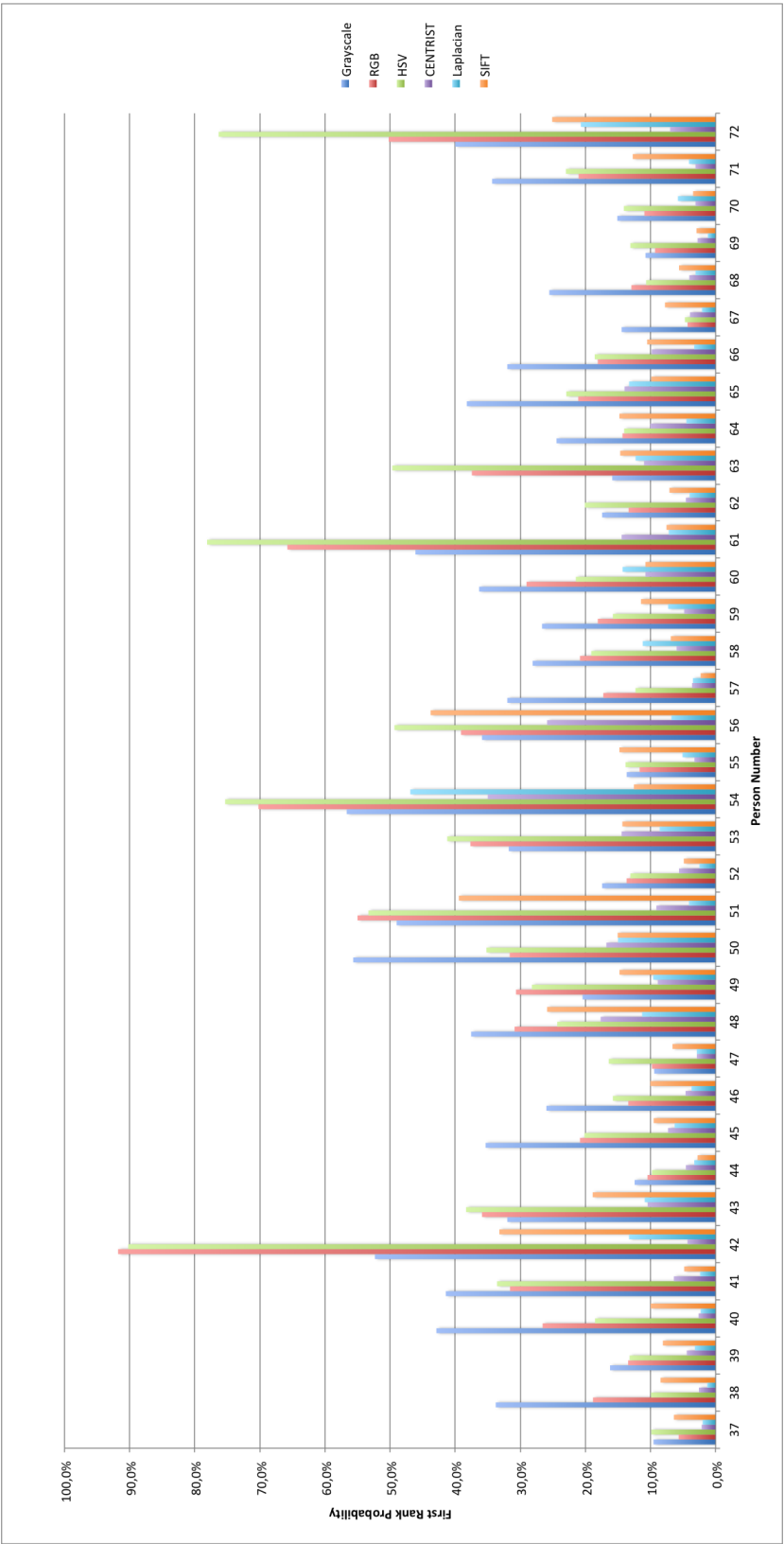


Figure 4.34: 1st Rank Result for Each Person and Each Method

Chapter 5

Proposed Appearance Model

The structure of a generic appearance model starts by selecting the most appropriate features to extract from the person and defining how to measure the similarity between input persons and the available models, which is followed in this proposal. It is then expanded into a resolution driven model, which uses the image resolution to select the appropriate action and then turned into a 3D Model which is used to improve re-identifications even when the pose isn't directly available.

5.1 Initial Model Overview

An overview of the initial proposed model is presented in Figure 5.1, which includes three color features: grayscale histograms, RGB histograms and HSV histograms as well as two texture features: CENTRIST histograms and Laplacian Histograms. Since comparisons are made with histograms with different bin numbers, an additional normalization step is required. For the histograms, since the previously used Chi-Square distance is not normalized, the comparison metric will be the state of the art Bhattacharyya Distance, which provides normalized values from 0 (equality) and 1 (completely different) [124].

The result on the appearance model then relies on simple sums and multiplications of the results of histogram comparison with weights w_1 to w_7 which will be experimentally determined in the next section. The tree structure will allow optimizing the results within the type of feature before the weights given to the type of feature.

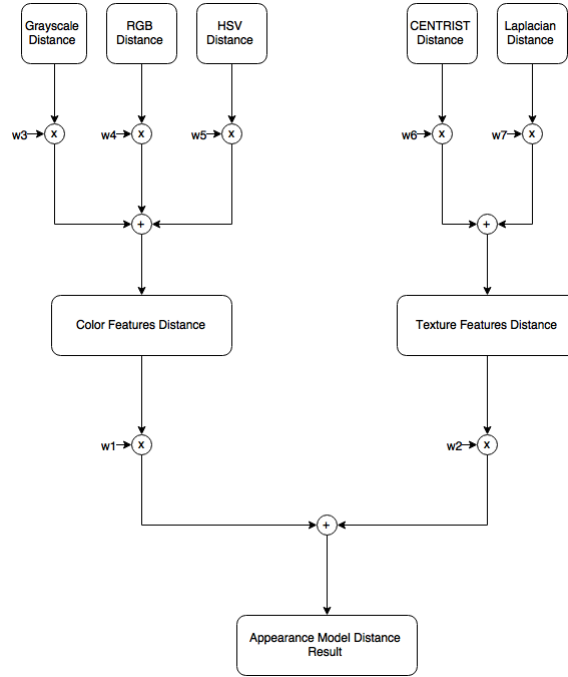


Figure 5.1: Overview of the Appearance Model. Values w_{1-7} represent the weights of the subparts of the model. Each feature / level must be normalized.

5.2 Best Weight Combination

The first step in optimizing the proposed model is to determine the weights w_{1-7} from Figure 5.1. These weights are determined experimentally from the top-down.

Starting with color features, a test to determine the values of w_3 , w_4 and w_5 was conducted. These values give the maximizing results for Global Color Features and were obtained by testing values from 0 to 1 in 0.01 increments for each weight. The maximizing weights are found in Equation 5.1. While w_3 is low, the grayscale histograms are maintained because they do provide an increase in the 1st rank and their memory occupation and processing time is negligible. The reason behind this may be because of the redundancy between these features and the use of the value channel of the HSV histogram. The CMC curves with the comparisons of the final color formulation and the individual color features is presented in Figure 5.2, which shows a 1st rank of 21.9%, just a decimal point above the use of the HSV histograms. However, even though their starting position is about the same, the combination of features gives better results on the higher ranks.

$$\begin{cases} w_3 = 4\% \\ w_4 = 56\% \\ w_5 = 40\% \end{cases} \quad (5.1)$$

To determine the values of w_6 and w_7 , a test was conducted: A random image from each person was chosen as the model and the remainder were used as input for the appearance model.

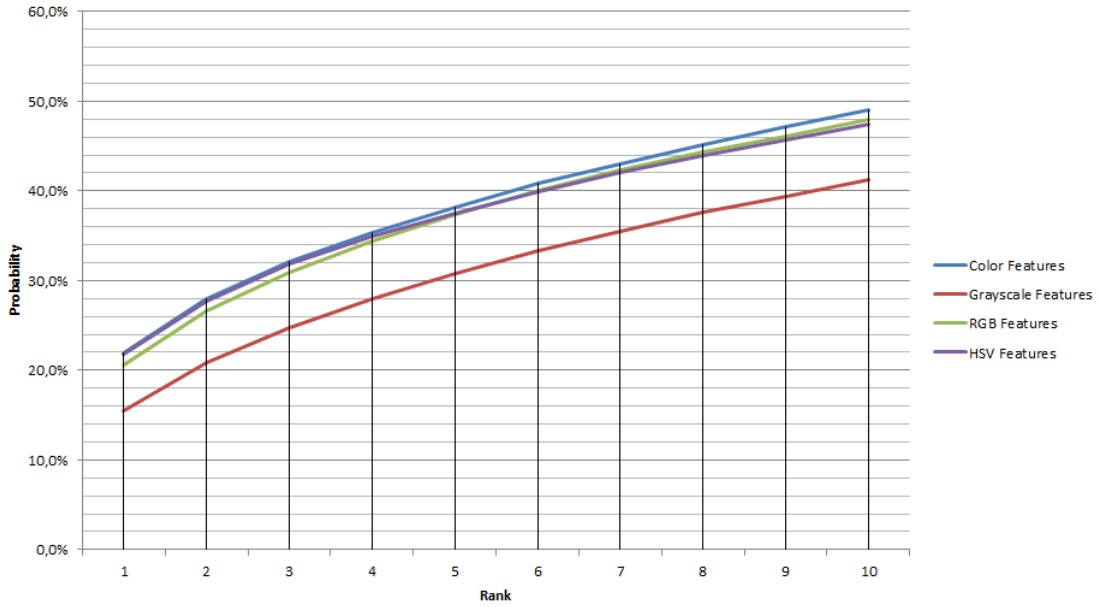


Figure 5.2: CMC Curve for Color Features

This procedure was repeated 100 times for each w_6 and w_7 pair to minimize the random choice of the model. The value of w_6 is tested from 0 to 1 in 0.01 increments while $w_7 = (1 - w_6)$. This will give a probability distribution for the Global Texture Features. The maximizing weights are presented in Equation 5.2, which shows that the texture features effectively work together as the best value doesn't occur when one of the features has a weight of 0.

$$\begin{cases} w_6 = 56\% \\ w_7 = 44\% \end{cases} \quad (5.2)$$

For this weight combination, the CMC curve of the texture features as well as the previously presented CENTRIST Histograms and Laplacian Histograms curves are shown in Figure 5.3, with a 1st rank result of 10.5%. This is an improvement on using the CENTRIST or the Laplacian individually, which confirms that they extract different information that can be used together to improve the results.

Finally, using the previously determined weights for w_{3-7} from Equations 5.2 and 5.1, the weights of w_1 and w_2 were determined. The maximizing weights are found in Equation 5.3, which shows that the best result is achieved when texture and color are given just about the same weight and reinforces the use of both types of features. In Figure 5.4, a CMC Curve of both features together and color and texture features is presented. The mixed model's curve is higher than the color features, with a 1st rank of 22.5%.

$$\begin{cases} w_1 = 44\% \\ w_2 = 56\% \end{cases} \quad (5.3)$$

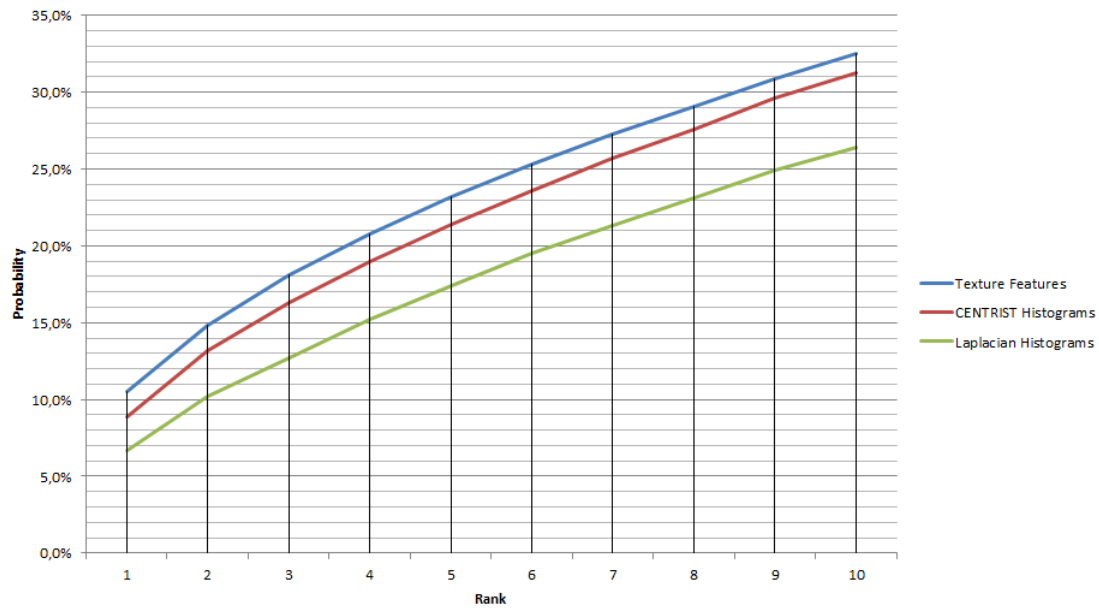


Figure 5.3: CMC Curve for Texture Features

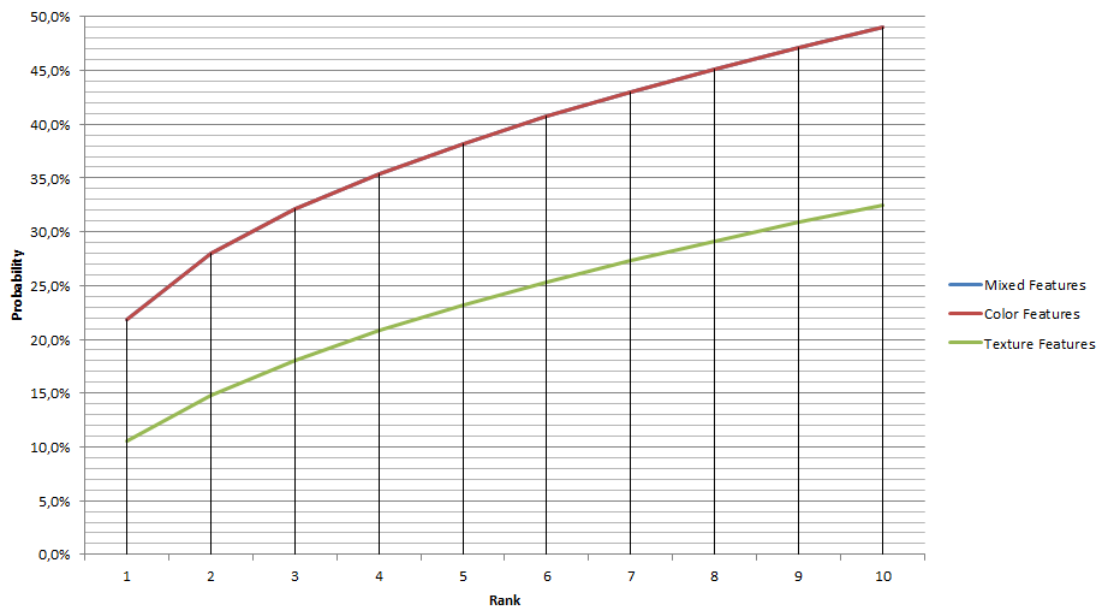


Figure 5.4: CMC Curve for Texture Features, Color Features and Mixed Model. The Mixed Model and Color Features almost overlay

5.3 Local Features

The addition of local features is dependent on the balance between performance and results. The extraction and comparison procedure of local features is more time consuming when compared to extracting and comparing color and texture histograms and as such, using local features in a real-time system is often not possible. However, and since the FAST detection brought the best results in previous tests, an extension of the model is proposed in Figure 5.5. In this case, a new weight is added at w_8 , which causes the need to redetermine w_1 and w_2 . The best set of weights is presented in Equation 5.4, which gives minimal weight on the local features.

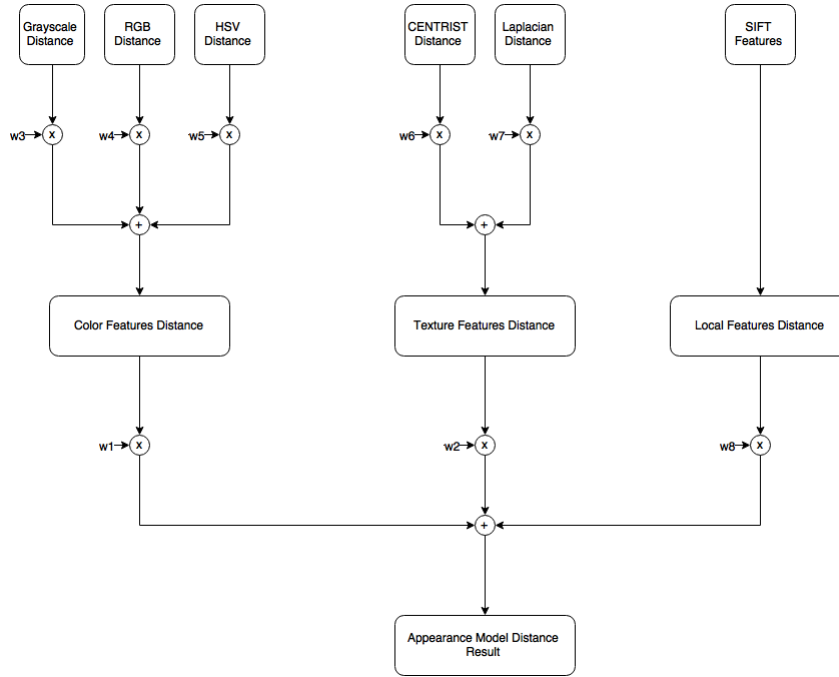


Figure 5.5: Overview of the Appearance Model. Values w_1-8 represent the weights of the subparts of the model. Each feature / level must be normalized.

$$\begin{cases} w_1 = 15\% \\ w_2 = 80\% \\ w_8 = 5\% \end{cases} \quad (5.4)$$

The 1st rank result isn't an improvement over using just color and texture features, as the re-identification rate is 22.1%, and it adds processing time due to the detection and extraction of features. Since there was no improvement in performance, local features won't be part of the model. The CMC curve can be seen in Figure 5.6.

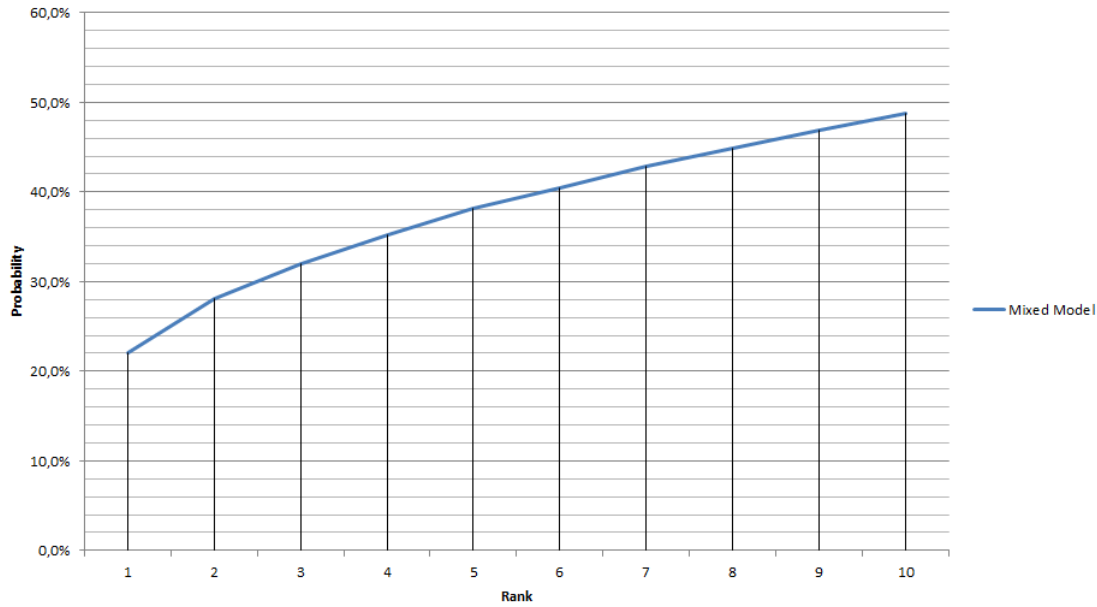


Figure 5.6: CMC Curve For Texture, Color and Local Features

5.4 Resolution Driven Appearance Model

While in an ideal situation only high resolution images are used, in a real scenario this is often not the case. When extracting information from low resolution images, this information is different; for example, low resolution images will have less texture information, as texture information is usually in the image details. A lower resolution also means that the images are more sensitive to noise. Extracting local features on these images would also prove to be inefficient as very little keypoints can be extracted. As such, the appearance model should make use of this information to be improved and therefore a change in it's structure is proposed.

5.4.1 Model Structure

The difference in the Resolution Driven Appearance Model is that instead of having a single model for each person, two models can be created: one for low resolution models and one for high resolution models. It was considered that when the person's region of interest was bigger than $[35 \times 70]$, the model has high resolution. The appearance model can be seen in Figure 5.7 and shows the appearance model structure. The values for the Color Feature Distance and Texture Feature Distance still follow the weights and structure from Figure 5.1.

Depending on the image that is used to create the model, two situations can occur: when the image is of high resolution, the high resolution model is extracted and the image is resized and the low resolution model is created. Resizing the image removes the detail information from the high resolution model, which is useful when comparing to other low resolution images; when the image is of low resolution, only the low resolution model is created. A schematic of the procedure is shown in Figure 5.8. This means that depending on the input image, one or two parts of the

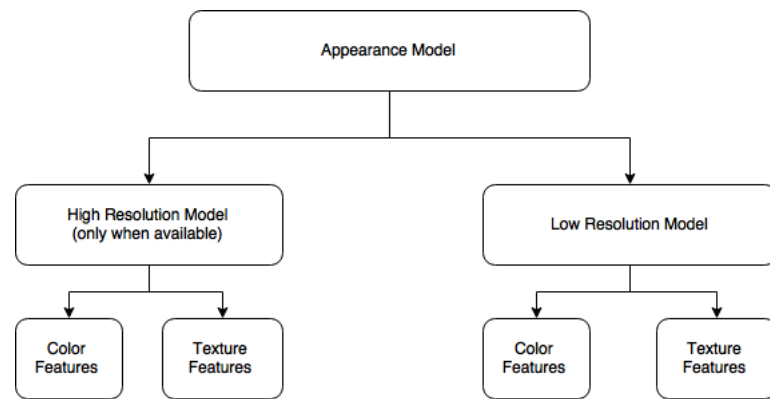


Figure 5.7: Appearance Model Extension, with High and Low Resolution

model are created. The steps of "Low Resolution Model" and "High Resolution Model" represent the creation of the models based on the Color and Texture features.

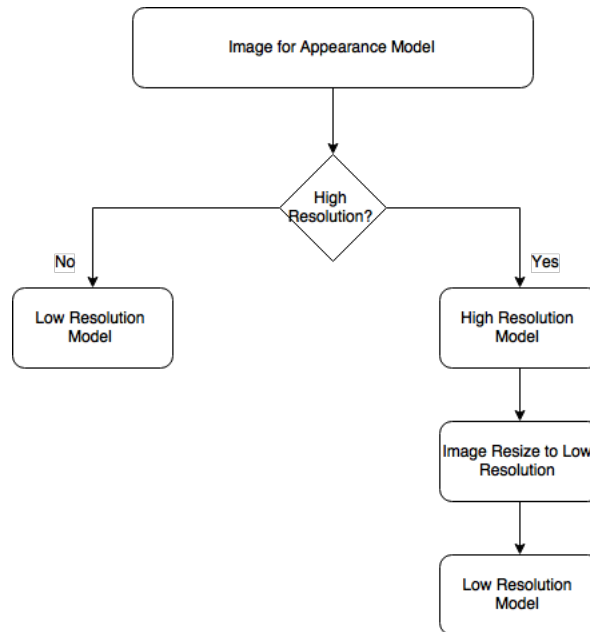


Figure 5.8: Model Creation Process

5.4.2 Model Comparison

Unlike texture information, resolution doesn't affect the extraction of color information to the same degree as it's usually something that's spread over the image and not in the details. When extracting texture information from a low resolution image, very little texture information is captured, but it can be used with other low resolution images. The process of comparing images to the model is shown in Figure 5.9. While this seems to add complexity, it's a simple decision tree with additional weights. The best selection of weights is found in Equation 5.5, which means that

in the lower resolution model, most of the weight is given to color while when the high-resolution model is available, they share the same weight.

$$\begin{cases} w_9 = 50\% \\ w_{10} = 50\% \\ w_{11} = 25\% \\ w_{12} = 75\% \end{cases} \quad (5.5)$$

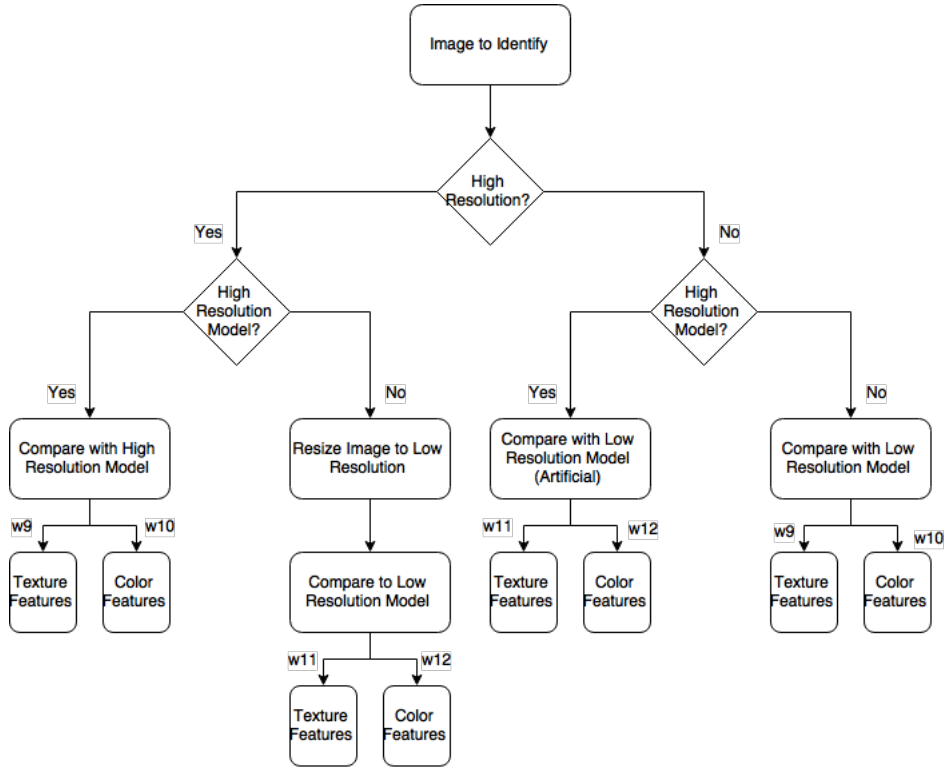


Figure 5.9: Resolution Driven Comparison Weights

5.4.3 Results

To test this change, one random image from each person is used as a model. If the resolution is below $[35 \times 70]$, the model or image is categorized as low resolution. A random image from each person is chosen as the model and the remaining images are used as comparison, with the weights reflecting their resolutions.

These values result in a 1st rank of 23.2%, 0.7% higher than the initial model, but this change comes at a negligible processing and memory cost, and will be kept. The CMC curve for this variation can be seen in Figure 5.10.

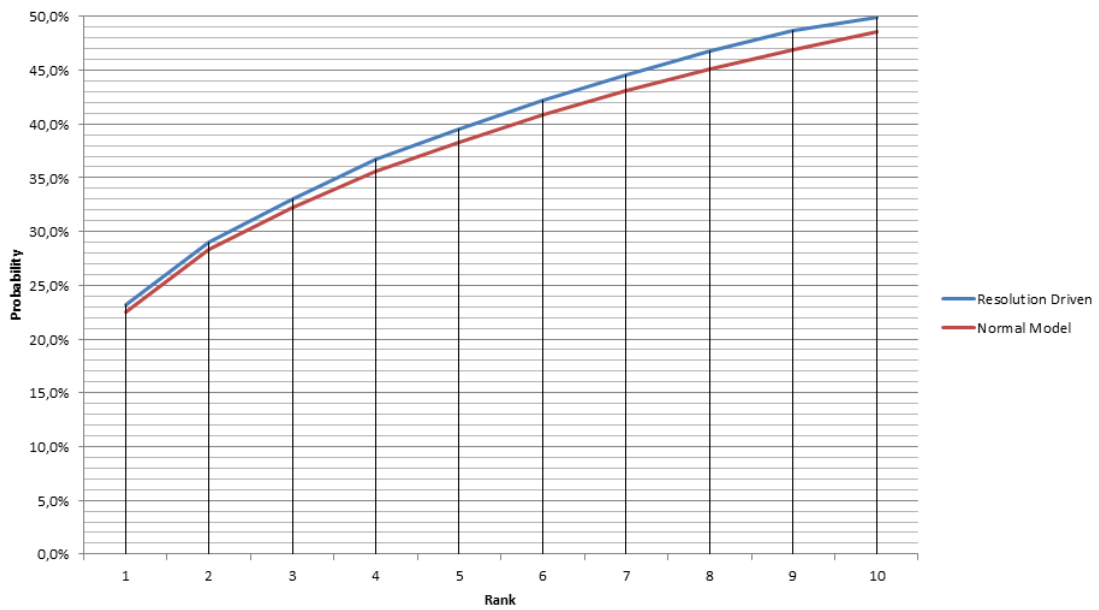


Figure 5.10: CMC Curve For Texture and Color Features with Resolution Driven Comparison Weights

5.5 Multi-Dimension Model

When a person appears on camera, it can show a wide variety of different poses with completely different appearances. In Figure 5.11, an example of a person from the CAVIAR4REID dataset is shown with different viewing angles. In this case, the person can be seen facing the camera, with his back turned to the camera and moving sideways. These views show several differences: the front/back views are widely different in the head area and the side-views differ from the front/back because of the addition of a big portion of background.



Figure 5.11: Different Poses for a Single Person in the CAVIAR4REID Dataset

With this in mind, the idea is that instead of having a single model for the person, a three body part model is created: a front view, a back view and a side view, which balances the number of poses (which shouldn't be too high because of memory cost and processing time on comparisons) with the different ways the person can appear.

5.5.1 Pose Identification

To manage three poses of the model, the person's pose must be correctly classified as otherwise too much noise would be added to the model. This information may be given by the tracking algorithm or though a pose detection algorithm, which evaluates the movement of the person. The tracking algorithm is capable of storing a trajectory of the person by using the consecutive points in which a person is found. This information can be used to estimate the direction that the person is taking: if the bounding box is moving up, the person is moving away from the camera (back view); if the bounding box is moving down, the person is moving towards the camera (front view); if the bounding box is moving sideways, the person is moving sideways (side view).

This requires that a delay is added before the tracking algorithm requests a re-identification. The pose is estimated over a bi-directional window with the initial point, P_i being defined as in Equation 5.6 and the final point, P_f being defined in Equation 5.7. Here, Min represents the oldest available point from $P_{frame-window}$ to P_{frame} and Max represents the most recent available point from P_{frame} and $P_{frame+window}$. Equation 5.8 defines how to determine the angle. The proposed window size is the frame equivalent to half a second.

$$P_i = Min(P_{frame-window} \text{ to } P_{frame}) \quad (5.6)$$

$$P_f = Max(P_{frame} \text{ to } P_{frame+window}) \quad (5.7)$$

$$Angle = atan_2\left(\frac{P_{f,y} - P_{i,y}}{P_{f,x} - P_{i,x}}\right) \quad (5.8)$$

It should be noted, however, that this formulation is valid considering the camera configuration present in the CAVIAR dataset. Adaptations to other cases would be needed.

5.5.2 Results

To test the results, ground-truth data for the pose will be used. Each of the images in the dataset has been annotated according to the direction: facing camera, back turned to camera and sideways.

The first test will select a random image for each of the 72 persons to be used as a reference model for that pose. The remaining images will, when they have a model with that perspective in the system, be used for testing. This allows testing if re-identifying images when the pose is well known and a model with that pose is present improves the re-identification. A 1st rank result of 40.5% is achieved, which almost doubles the previous result of 23.2%. The CMC curve can be seen in Figure 5.12. It does, however, include a reduced number of comparisons.

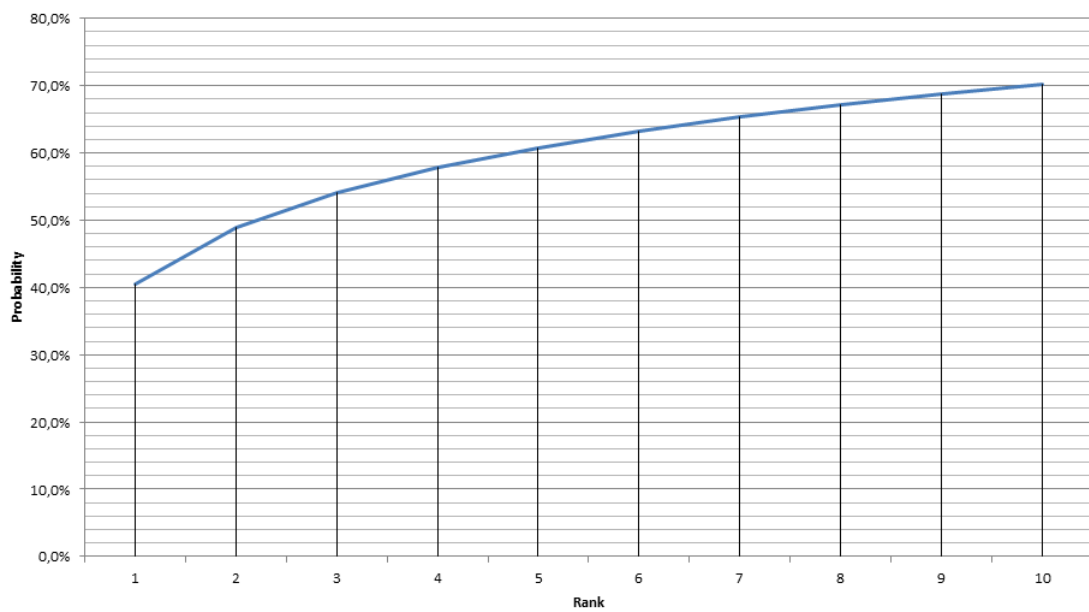


Figure 5.12: CMC Curve with Pose Ground-Truth and Available Learned Pose (Single Pose)

Instead of comparing to every model, the next test compares to all models, which results in a 1st rank result of 45.3%, with the CMC curve of Figure 5.13. In this case, again, the pose must be available as part of a model in the system for the image to be tested.

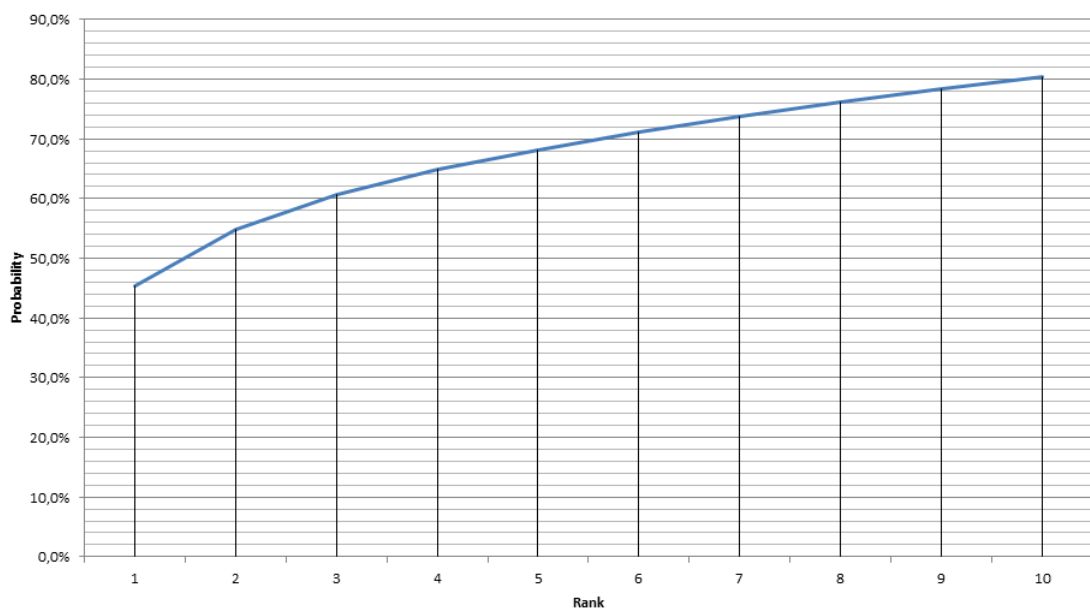


Figure 5.13: CMC Curve with Pose Ground-Truth, Available Learned Pose and Comparisons to Same Pose Only

When a model is built for each person and all images are used for comparison, but comparisons are made with models with the same pose only, the resulting 1st rank is 21.7% with the CMC curve

of Figure 5.14. These results, very similar to those not using any pose verifications, suggest that most of the previously correct matches were being made on the same pose results. In fact, 93.5% of correct results come from re-identifications made to the same pose.

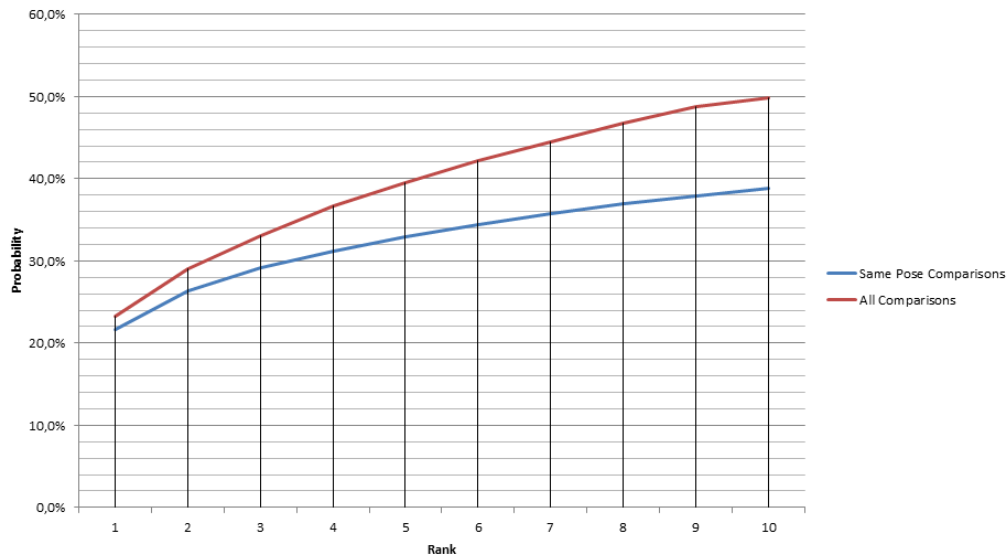


Figure 5.14: CMC Curve with Pose Ground-Truth and Comparisons to Same Pose Only

To determine whether having more than one model per person improves the re-identification results, one model is built for each available pair of pose and person, with comparisons made with models with the same pose only. This results in a 1st rank of 38.0% and the CMC curve of Figure 5.15. This shows the improvement that multi-shot models can have, in particular when they take advantage of the pose information.

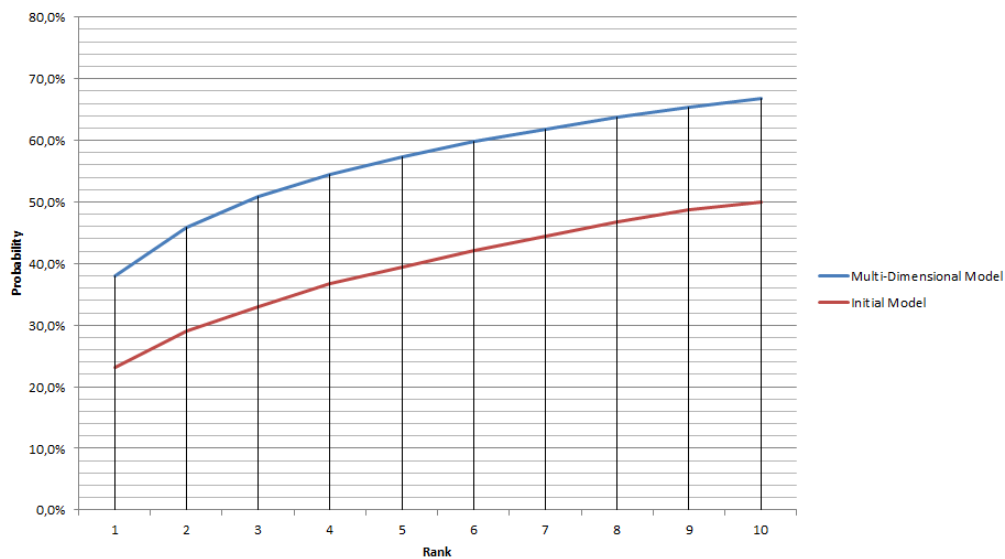


Figure 5.15: CMC curve with Pose Ground-Truth, All Pose Models and Comparisons to Same Pose Only

5.6 3D Model Extension

A natural extension to the multi-dimensional model is the 3D Model. A person's pose isn't always facing camera, sideways or with back turned to the camera, but a multitude of other possible angles. As seen in the previous section, most correct matches come from having the same pose of the model in the system. The idea behind this extension is to use the information from available poses (front, back or side) and infer other poses. The objective is to approximate the person's appearance model to the one in Figure 5.16 with a generic angle. Only front, side and back poses are determined based on real images of the person and the remaining poses are interpolated.

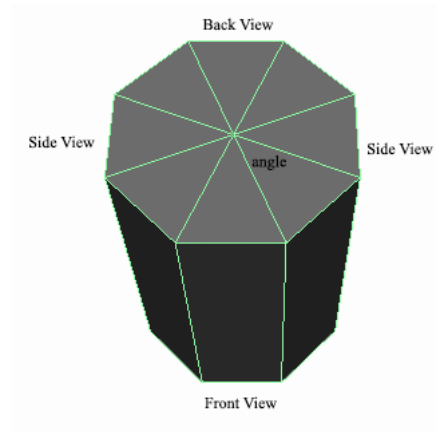


Figure 5.16: Generic 3D Appearance Model: Instead of using 3 views, the model can adapt to a generic n views

5.6.1 Interpolation Algorithm

Without information of the model from all the angles, a good interpolation algorithm needs to be used. Considering the spacing, in degrees, between the poses as *angle* and with n models to interpolate, the pseudo-code is presented in Algorithm 1. The idea is that since the features are stored in histograms, an interpolation of the desired extra positions is made by using a weighted average. The normalization step afterwards ensures that the histograms remain normalized for comparison.

5.6.2 Angle Variations

Previous tests with a multi-pose model have shown re-identification rates of 38.0% even before interpolation. To properly test the results of the 3D model, the tests detailed in Table 5.1 will change how differences in the angle and interpolation affects the re-identification. In all tests, all available poses are built as part of the model and comparisons are only made when the testing pose is available.

Algorithm 1: Interpolation Algorithm for Creation of Models

Data: Histograms for Poses at $angle_1$ and $angle_2$, with $angle \leftarrow angle_2 - angle_1$
Result: n Histograms, spaced $\frac{angle}{n}$
 $currentAngle \leftarrow angle_1 + \frac{angle}{n}$;
 $originalHistogram \leftarrow Histograms[angle_1]$;
 $finalHistogram \leftarrow Histograms[angle_2]$;
 $iteration \leftarrow 1$;
while $iteration < n$ **do**
 $newHistogram \leftarrow \frac{iteration}{n} \times originalHistogram + \frac{n - iteration}{n} \times finalHistogram$;
 $newHistogram \leftarrow \text{normalize}(newHistogram)$;
 $Histograms[angle_1 + iteration \times angle] \leftarrow newHistogram$;
 $iteration \leftarrow iteration + 1$

Angle	Comparison Pre-Requisites	Test Objective	1 st Rank	Figure
45°	Comparison is only made with matched pose	Verify if the interpolated models achieve good results	36.7%	5.17
45°	Comparison is made with matched pose and 1 adjacent	Verify if comparing to neighbors increases the results	39.9%	5.18
45°	Comparison is made with all models	Verify if extra poses introduce noise	38.8%	5.19
30°	Comparison is made with matched pose and 1 adjacent	Verify if a smaller angle is better	39.0%	5.20
30°	Comparison is made with matched pose and 2 adjacents	Verify if a wider comparison is better	41.5%	5.21
22.5°	Comparison is made with matched pose and 3 adjacents	Verify if a smaller angle is better	41.9%	5.22
18°	Comparison is made with matched pose and 4 adjacents	Verify if a smaller angle is better	41.9%	5.23
15°	Comparison is made with matched pose and 5 adjacents	Verify if a smaller angle is better	41.4%	5.24

Table 5.1: Testing Overview for the 3D Model

In the first test, the image is only compared when there is a match in the pose and that pose is available. This leads to a 1st rank result of 36.7%. The respective CMC curve is Figure 5.17. It shows that even when only using the interpolated model the results are close when compared to the pose model but with all images being compared.

When the comparison is extended to comparisons on not only the matching pose but also adjacent poses, the re-identification rate increases to 39.9% with the CMC curve of 5.18. This result is already superior when compared to the pose model but all images are now being used for comparison.

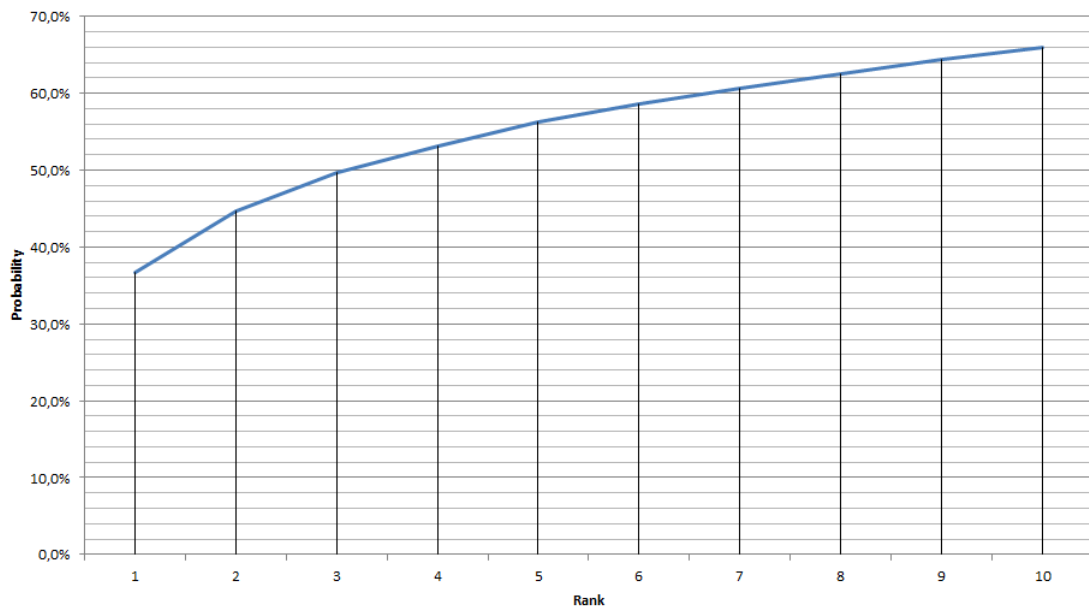


Figure 5.17: CMC Curve with Pose Ground-Truth, Interpolated Missing Poses, Comparisons to Pose and When Pose is Available

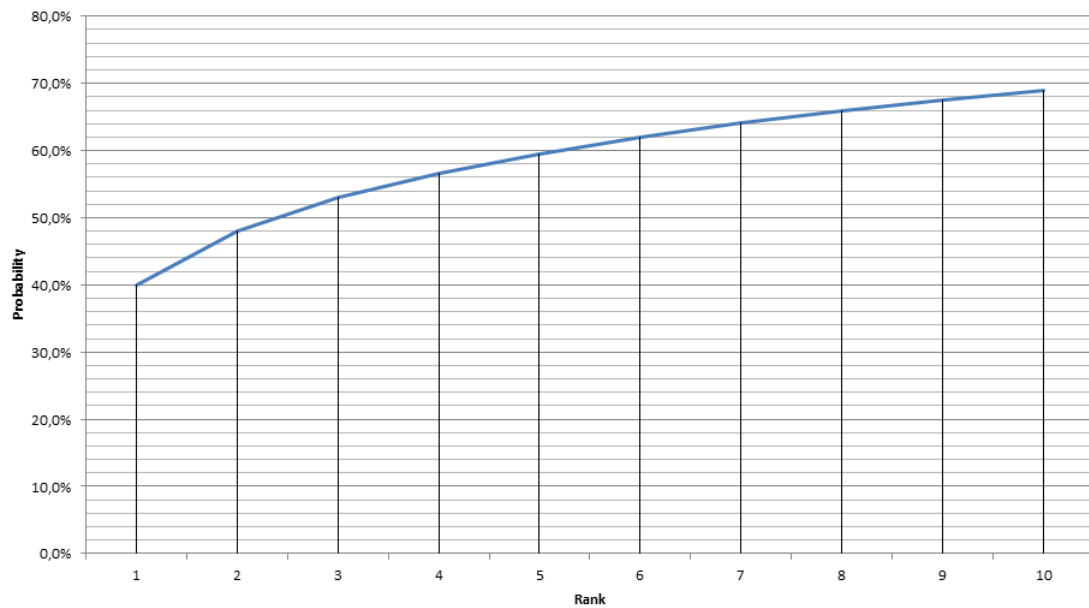


Figure 5.18: CMC Curve with Pose Ground-Truth, Interpolated Missing Poses, Comparisons to Pose and When Pose or Adjacent Pose is Available

To test if the increasing number of models reduced the re-identification rate when comparisons are made to all poses, a test under these conditions was made, which resulted in a 1st rank result of 38.8%. This means that the addition of poses won't produce significant noise even when the pose isn't verified. The respective CMC curve is [Figure 5.19](#).

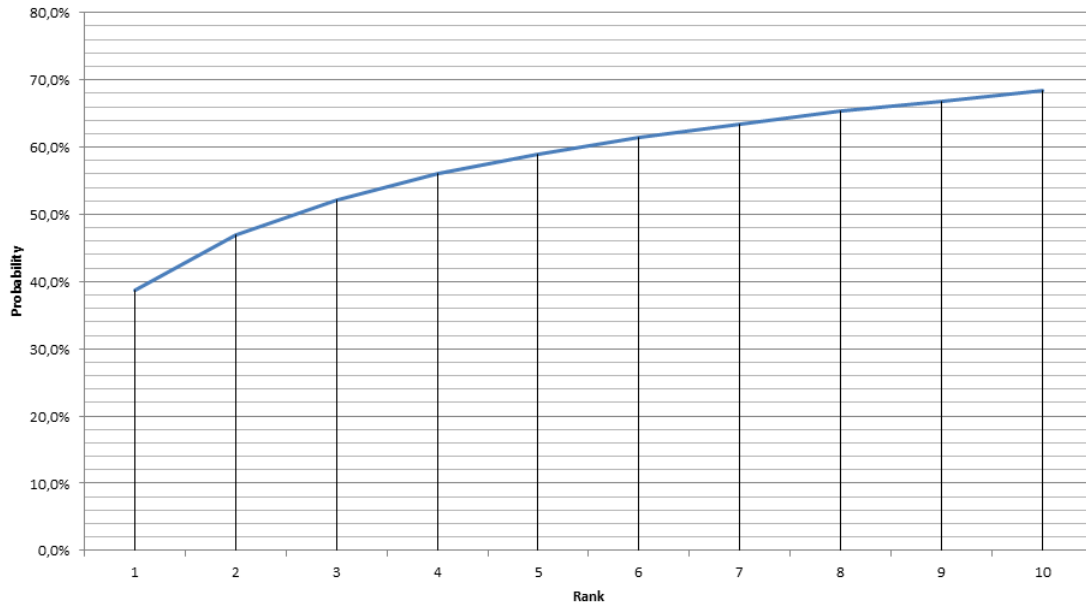


Figure 5.19: CMC Curve with Pose Ground-Truth, Interpolated Missing Poses, Comparisons to All Poses

Until this point, only tests under an angle of 45° were made. The algorithm is flexible when dealing with different angles. The first tested alternative angle is 30° . Comparisons are made with the ground-truth pose and adjacent poses, which brought the best results in the previous scenario. The 1st rank result of 39.0% is achieved. The CMC curve is seen in Figure 5.20.

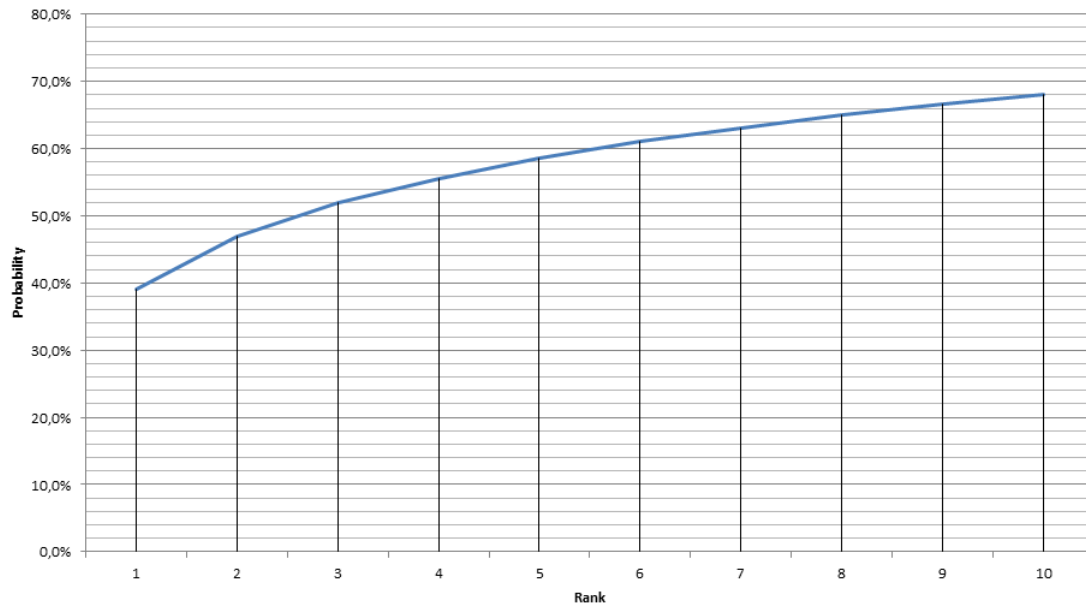


Figure 5.20: CMC Curve with Pose Ground-Truth, Interpolated Missing Poses, Comparisons to Same Pose and Adjacents

To decrease the uncertainty from the small variations of angle, images are now compared to

the same pose as well as the two adjacent poses. This leads to a 1st rank result of 41.5%, which is an improvement over the previous case. The CMC curve is seen in Figure 5.21.

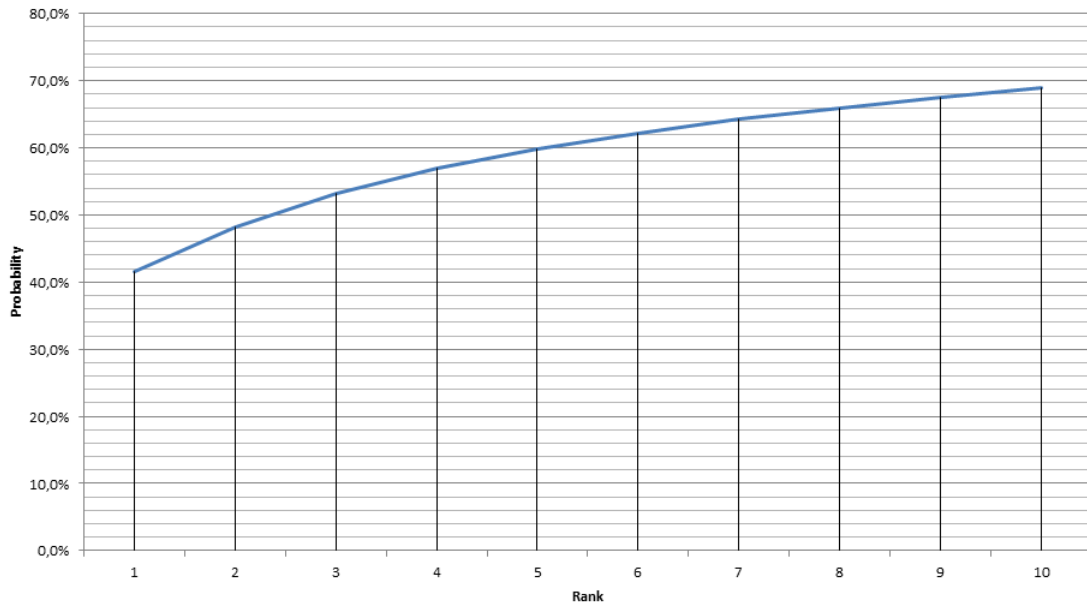


Figure 5.21: CMC Curve with Pose Ground-Truth, Interpolated Missing Poses, Comparisons to Same Pose and 2 Adjacent

The next angle variation used 22.5° . With it, comparisons with ground-truth pose and 3 adjacent poses are made which maximized the 1st rank result to 41.9% and the CMC curve of Figure 5.22.

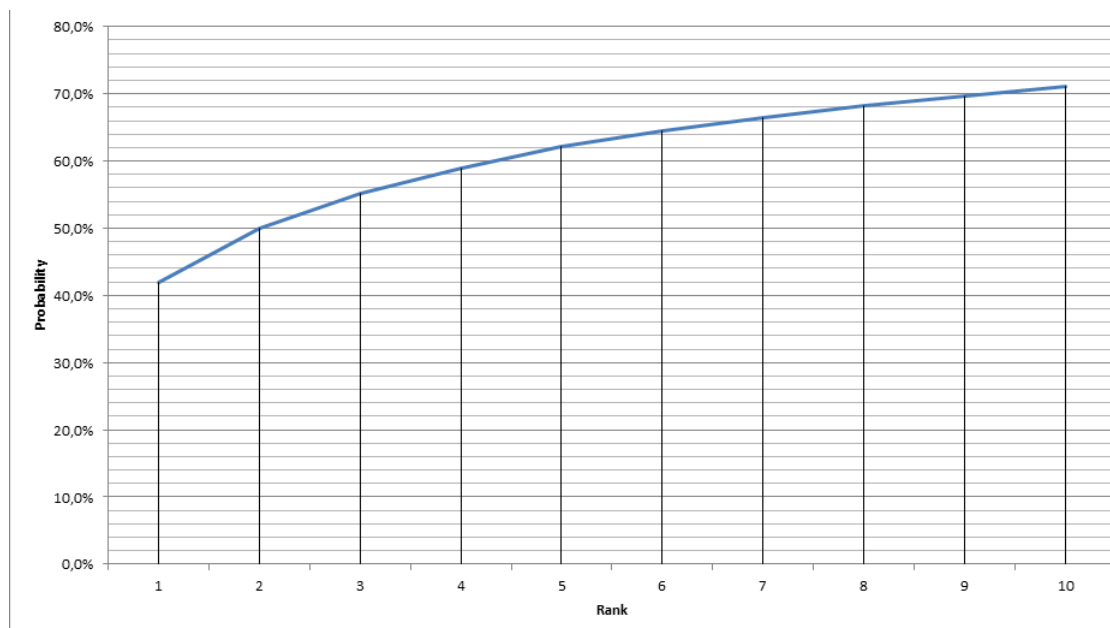


Figure 5.22: CMC Curve with Pose Ground-Truth, Interpolated Missing Poses, Comparisons to Same Pose and 3 Adjacent

When testing angles of 18° and 15° , the best result was 41.9% with the CMC of Figure 5.23 and 41.4% with CMC of Figure 5.24, respectively. This shows that an increase in the number of models will result in a lower re-identification rate.

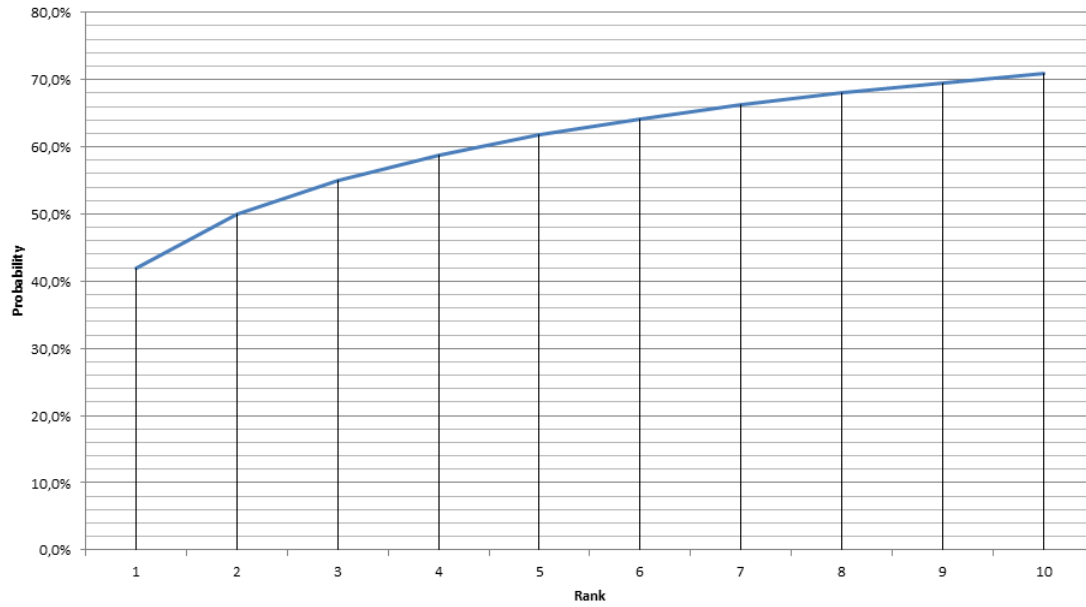


Figure 5.23: CMC Curve with Pose Ground-Truth, Interpolated Missing Poses, Comparisons to Same Pose and 4 Adjacents

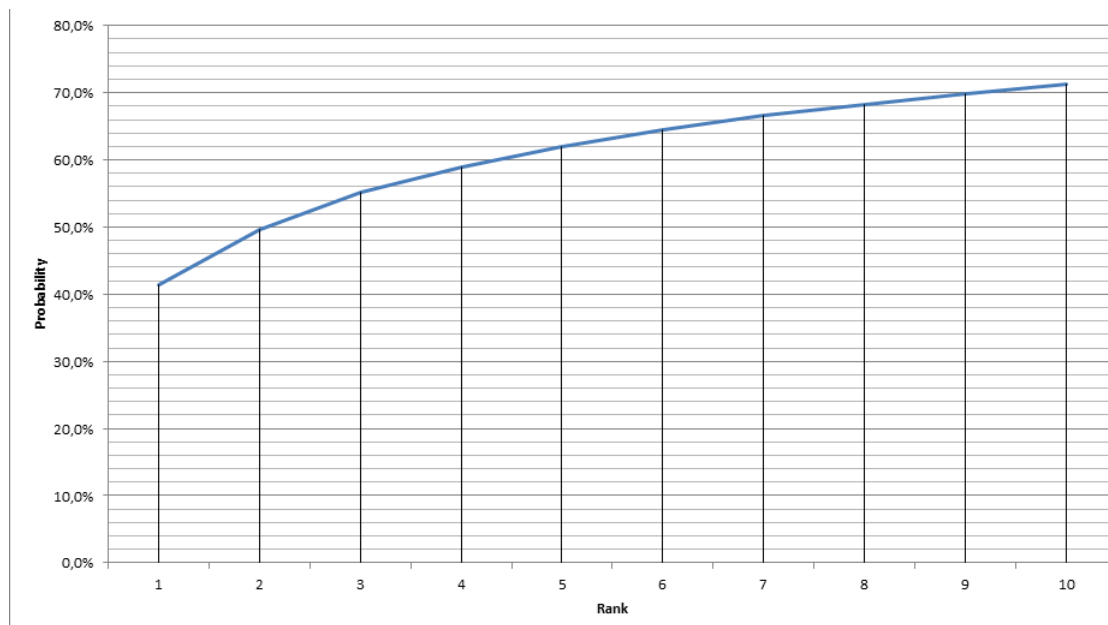


Figure 5.24: CMC Curve with Pose Ground-Truth, Interpolated Missing Poses, Comparisons to Same Pose and 5 Adjacents

These tests show that the use of the 3D Model can improve the re-identification results. On

a single-shot model, the re-identification rate was 23.2% but with the 3D Model, it increases to 41.9%. Even comparing to an ideal case in the previous multi-resolution model, where a model is created for each pose only (without any interpolation of the remaining poses) the result of the 3D model is better (as the multi-resolution algorithm only reaches 38.0%).

5.7 Low Resolution Problems

The problem with the use of the CAVIAR4REID Dataset is that there are many very low resolution images, with some examples shown in Figure 5.25. The problem with these images is that they often have very little detail and are subject to noise, which creates challenges in model creation and re-identification, with some of them being hard for a human to manually identify. To verify how much of an improvement the system would get if only high resolution images were used, a comparison will be made with the best 3D Model from before, which had a 1st rank result of 41.9%.



Figure 5.25: Example of Low Resolution Images used for Re-Identification or Appearance Model

A "high-resolution" image is considered to be when it is over 85 pixels tall. An example of an image that's 86 pixels tall is shown in Figure 5.26. While it's not clear of noise, its dimensions allow for enough detail for the person to be identified by a human with relative ease.



Figure 5.26: Example of a High Resolution Image

In this test, since most images are below those dimensions, only about 270 re-identifications attempts are made (on a dataset with 1220 images), even though all the models are created. This is because of the large amount of low resolution images in the dataset. This experiment has lead to a 1st rank result of 72.5% and the CMC curve of Figure 5.27. Within the first 5 attempts, 90% of matches are made.

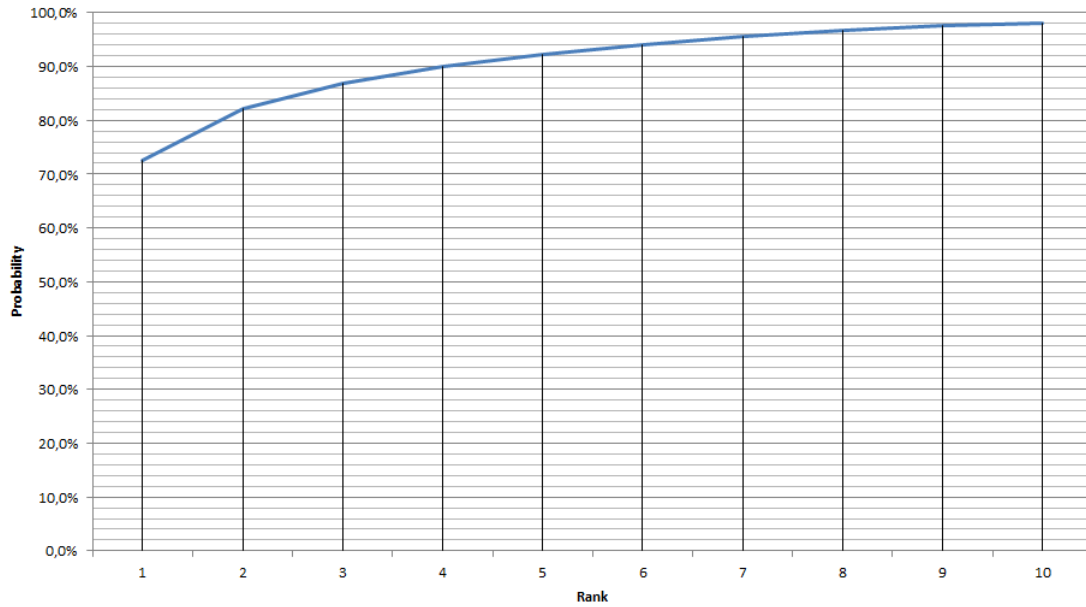


Figure 5.27: CMC Curve with Pose Ground-Truth, Interpolated Missing Poses, Comparisons to Same Pose and 3 Adjacent with High-Resolution Images Only

To test whether having high resolution models is enough to improve the performance, these models are created for the available poses and all other images are used as input for re-identification. This resulted in a 1st rank result of 53.3% and the CMC curve of Figure 5.28.

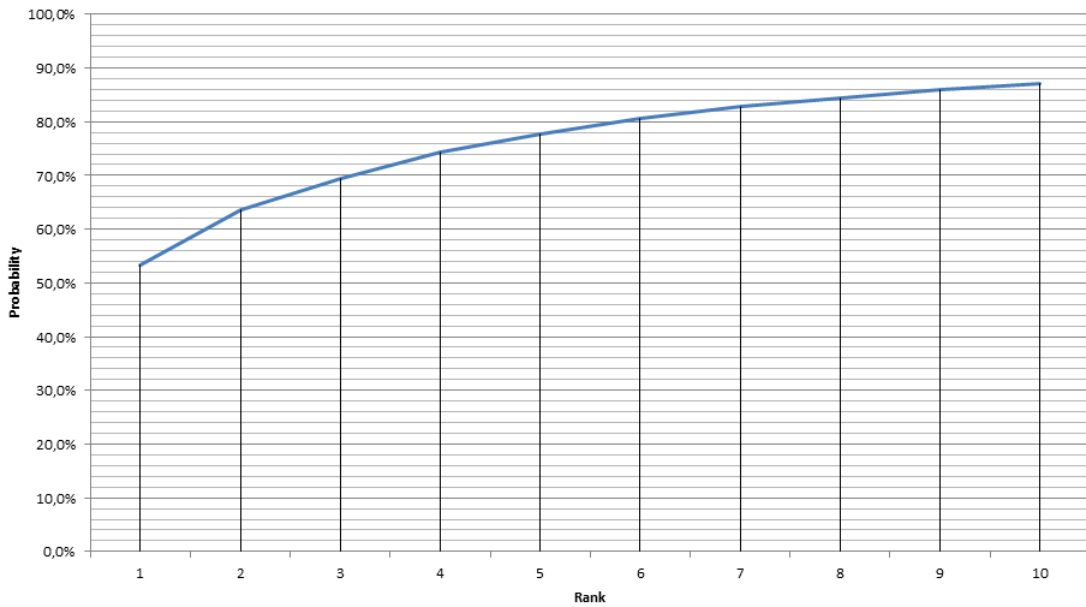


Figure 5.28: CMC Curve with Pose Ground-Truth, Interpolated Missing Poses, Comparisons to Same Pose and 3 Adjacent with High-Resolution Models Only

As seen in the testing results, the fact that the regions of interest in the dataset often have low resolution causes a big decrease in the re-identification results. In fact, when only high resolution

regions of interest are used, almost three quarters of matches are made on the first attempt, while just the use of high resolution models only already represents an improvement of over 10%.

5.8 Learning Model

Up until this point, the testing considered that the person would already have a model built in the system and that the only process that could occur is re-identification. However, on a real system, this is often not the case and the system will have to identify new persons in the scene as well as update their models according to how different they are with the stored models (a person, either physically or through changes in illumination / camera perspective, changes its appearance through time).

Since all histograms comparisons are normalized, a value from 0 to 1 is the natural output of a comparison between a model and the image of the person, with 0 being the closest and 1 the farthest. This means that a series of decisions can define the different system behavior possibilities.

5.8.1 Decision Tree

When a new person is requested for re-identification, it enters the appearance model module and a series of decisions will lead to the expected behavior. The proposed decision tree is seen in Figure 5.29, which can end with the creation of a model, re-identifying a person, updating the model and re-identifying or refuse to ID (when the certainty level is low).

To determine the threshold values, certain objectives are proposed. These are not the only possibility but can be used as a starting point in a real scenario.

- $Threshold_{NewModel}$ - For this threshold, the objective is that most new models are identified but the system shouldn't create unnecessary additional models. This can be measured with state of the art metrics *precision*, *accuracy* and *recall*.
- $Threshold_{HighConfidence}$ - For this threshold, the objective is that when a match is made, it is correctly identified in over 90% of the cases. This corresponds to a 1st rank result of at least 90%.
- $Threshold_{UpdateModel}$ - For this threshold, the objective is that when a match is made, it is correctly identified in over 75% of the cases. This corresponds to a 1st rank result of at least 75%.

5.8.2 $Threshold_{NewModel}$

To determine this threshold, all the images were tested: if they are below the threshold when compared to all available moments at that point, it is considered to be a model already in the system; otherwise a new model is created. For the *precision*, *accuracy* and *recall* metrics, there is a need to define a True Positive, False Positive and False Negative: a true positive occurs when the

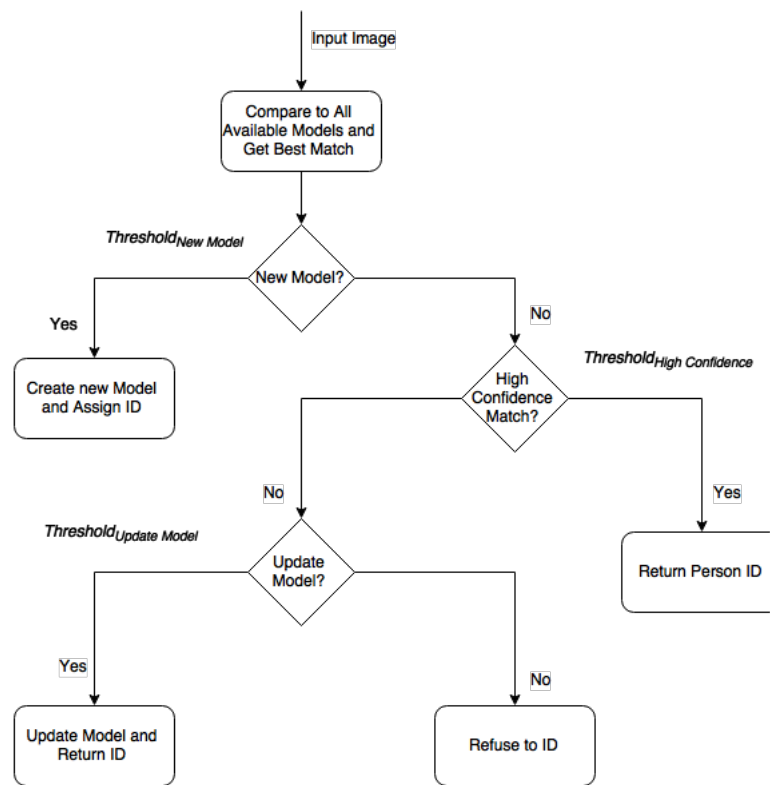


Figure 5.29: Appearance Model Decision Tree

system determines that it's a new model and no model with that ID is yet available in the system; a false positive occurs when the system determines that it's a new model but in fact it is not; a false negative occurs when the system determines that it is not a new model but in reality it is.

For each tested threshold value, 100 attempts are made, to minimize the random factor, and the *precision*, *accuracy* and *recall* were determined. The results are seen Figure 5.30. The threshold that's been chosen is 0.28, which has a *precision* of 91%. Even though the *recall* is just 29%, if the remaining thresholds are correctly chosen, those cases will be discarded with no ID and the tracking algorithm will request a re-identification in another frame.

5.8.3 $Threshold_{HighConfidence}$

To determine this threshold, the models were build beforehand and the matching is made with the remaining images, with the pose or adjacents. A re-identification is only given if the dissimilarity is low enough, which means that the models are very close in appearance. The objective is to maximize the number of matches while still getting a 90% re-identification. This means determining the highest threshold value with 90% re-identification.

The threshold is placed at 0.145, as it's close to the 90% mark. The first rank probability varies according to Figure 5.31.

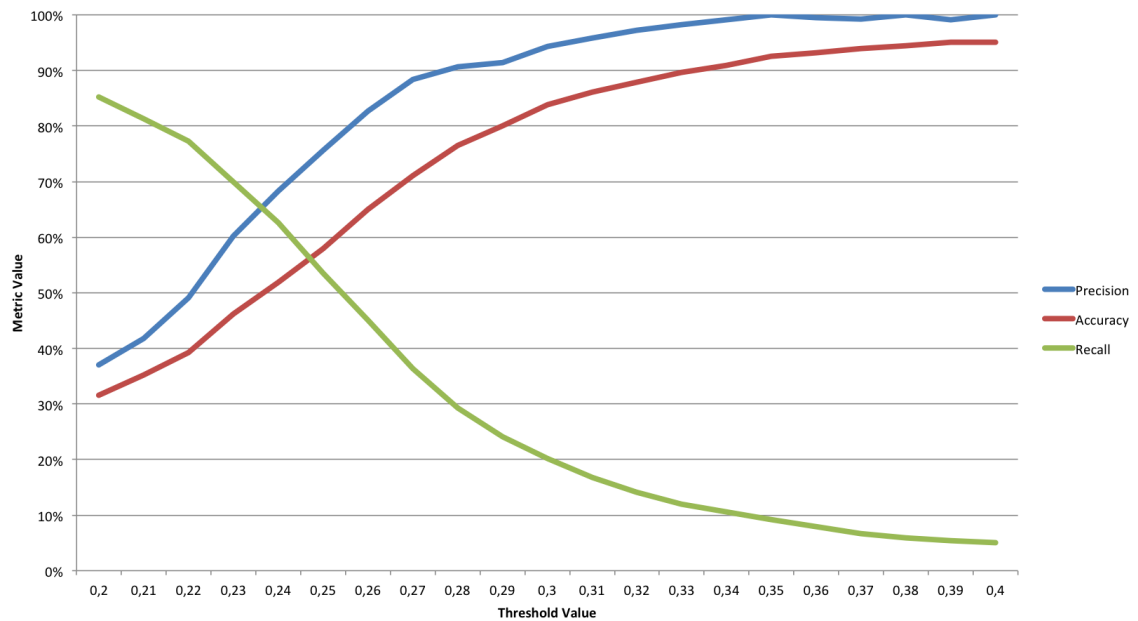


Figure 5.30: Precision, Accuracy and Recall for Threshold Values

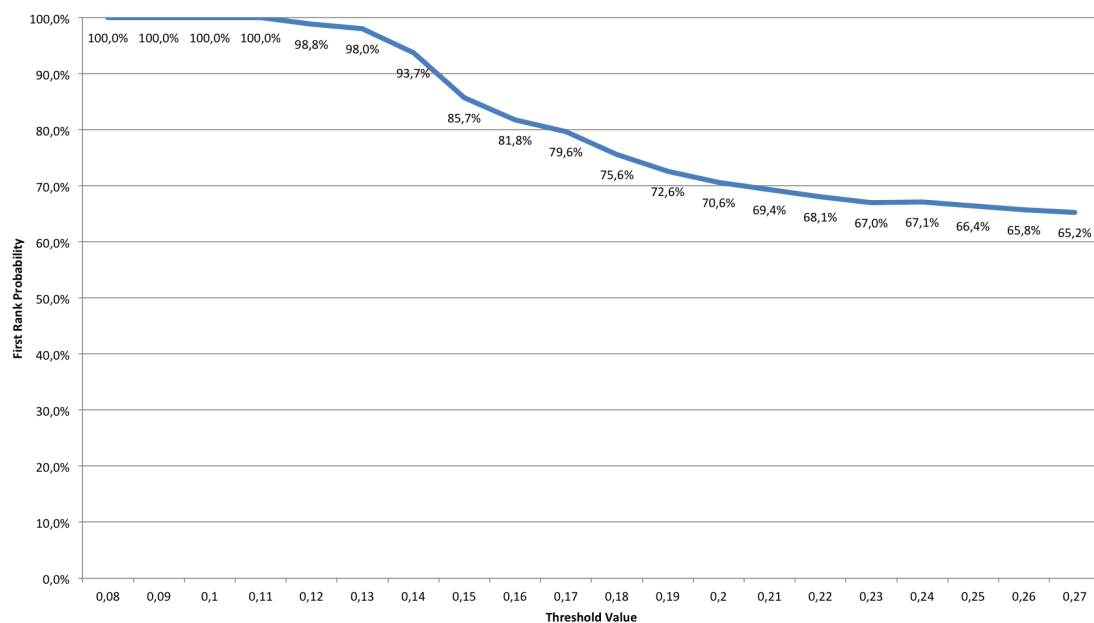


Figure 5.31: First Rank Probability of Re-Identification Ignoring Matches Above Threshold

5.8.4 $Threshold_{UpdateModel}$

Similarly, this threshold is placed at a first rank limit, in this case of 75%. This is to ensure that enough models are being updated, but that not too much noise is added. The idea behind updating

the model is that it adapts to gradual changes in the scene. However, there's no complete certainty that when a request is made to update the model that it's because the identified person has changed their appearance, which is why replacing the model isn't a viable option.

To balance both the need to update the model and not letting the model completely deviate from the person, updating the model will consist of averaging the model histogram with the new histogram.

The chosen threshold value is 0.18, which is close to crossing the 75% mark, as seen in Figure 5.31.

5.9 Appearance Model Results

To test the results of the appearance model, a comparison with other state of the art algorithms is needed. Since implementing these algorithms would be impossible in the available time frame, testing scenario of the 3D model is adapted to state of the art testing conditions.

In [42], a random subset of N images is selected for each person to build the gallery, with another N images being used for testing, with this procedure being repeated 20 times. While the original authors don't use the CAVIAR4REID Dataset, in the survey in [108], some CMC results are shown. A similar case is in [27], where single and multi-shot versions of the algorithm are compared. In [45], results for several values of N are presented. The results using a single model are compiled in Table 5.2.

Algorithm	1 st Rank
Multiple-shot Person Re-identification by Chromatic and Epitomic Analyses [45]	8%
Person re-identification by symmetry-driven accumulation of local features [42]	9%
Custom Pictorial Structures for Re-identification [27]	9%
Proposed Appearance Model	23.2%

Table 5.2: Result Comparison with Single-Shot State of the Art Algorithms on the CAVIAR4REID Dataset

Algorithm	1 st Rank
Person re-identification by symmetry-driven accumulation of local features (N = 5) [42]	17%
Custom Pictorial Structures for Re-identification (N=5) [27]	17%
Person re-identification by probabilistic relative distance comparison (N = 5) [44]	46%
Multiple-shot Person Re-identification by Chromatic and Epitomic Analyses (N=5) [45]	10%
PCCA: A new approach for distance learning from sparse pairwise constraints (N = 5) [49]	39%
Local fisher discriminant analysis for pedestrian re-identification (N = 5) [28]	37%
Multi-Shot Re-Identification with Random-Projection-Based Random Forests (N = 5) [50]	69%
Proposed Appearance Model (N = 5)	46%

Table 5.3: Result Comparison with Multiple-Shot State of the Art Algorithms on the CAVIAR4REID Dataset

When using multiple-shots of the person, some of these algorithms significantly improve their results. In Table 5.3, multi-shot algorithms are compared for the CAVIAR4REID dataset. The proposed appearance model outranks most state-of-the-art algorithms, only tying with [44] and under-performing against [50]. However, these are learning models, which have to be trained with a subset from the testing dataset and often suffer from over-fitting problems. Since there is no information on which persons were used to model, there is no way of fairly comparing the models as they may be chosen according to their performance. When comparing to non-learning models, the proposed model outranks the compared state-of-the-art algorithms. The rank values are either extracted from their original articles or from the overview from [50].

Another popular dataset for Re-Identification algorithms is the VIPeR Dataset. This dataset consists of 632 pairs of people captured from two angles. While this dataset is popular for single-shot algorithms, it disables the use of the pose and 3D Model Interpolation, which leads to significantly lower results. In this case, the behavior of a random algorithm would lead to results of 0.16%. The proposed algorithms reaches results of 4.0%, significantly lower than State-of-the-Art Algorithms [42] (20%), [125] (16%) or [44] (15%). However, since the focus of the work was building a multi-shot appearance model, this dataset is not appropriate for testing.

Chapter 6

Conclusions

6.1 Final Discussion

While extremely important for the performance of security algorithms, the re-identification problem is complex because of occlusions, changes in illumination and errors in the detection and tracking algorithms. The appearance model and the tracking algorithm should work together to ensure a maximization of the results: the appearance model gives the tracking algorithm the results of re-identification and the tracking algorithm may choose the best moments (where occlusion is minimal) to request re-identification or let the appearance model know that some persons cannot be matched as the tracking algorithm knows already where they are.

The structure of an appearance model can be divided into choosing the right characteristics to extract the model and compare a new entry with the available models. With that in mind, the dissertation can be split into four big sections: (1) test of the individual features to verify their individual results in a realistic re-identification scenario; (2) analyze how the most promising features work for each of the persons in the dataset; (3) combine the features into an appearance model; (4) extend the appearance model to improve the results.

The individual tested features used a selection of state of the art features such as color, texture and local features, to model and re-identify the persons. These features have shown different degrees of success, with color features reaching re-identification rates of 21.8%, while texture and local features have just reached rates of 8.9% and 10.5% respectively.

A deeper analysis of the most promising features has shown that their performance was dependent on the persons and several situations would cause confusion, for example, when several persons wore a similar color shirt and trousers. It has also shown that the different features had some cases where all features got poor results but on others, some of the features work better than others.

The proposed appearance model was tested first with just color and texture features and re-identification rates of 22.5%, an improvement over just using a single feature (HSV Histograms). Color features amounted to 21.9% and texture features 10.5%. Later, local features were added, but the results were lower, at 22.1%.

Regarding the proposed extensions, the resolution driven appearance model allows dealing with two appearance models depending on the resolution of the images used for comparison and for the model, which resulted in a 1st rank of 23.2%. Then, a three pose model has been added, in which the person is represented not only by a single model but a model for the front, one for the back and one for a side view, which resulted in a re-identification rate of 37.0%. This model was later extended to a 3D model, which interpolated the missing poses to predict how the person would be viewed in angles which were not directly available, which resulted in a re-identification rate of 41.9%. When comparing to other state-of-the-art solutions this appearance model outperformed or matched all non-trained models.

6.2 Future Work

While the objectives were fulfilled, during development of the dissertation, several ideas were not fully developed and would be part of future additions for the thesis.

One of the most important parts would be to test the appearance model in a real tracking scenario. This includes the integration of the appearance model working with a state-of-the-art tracking algorithm. This would result in a series of false positives and poor regions of interest for the re-identification algorithm to handle as current tracking algorithms still introduce errors. It would also include the definition of a proper protocol to ensure that the tracking algorithm and the appearance model work together to achieve the best results. Also related to the integration in a real system, several methods of background subtraction could be tested, which could reduce the noise in the scene. In a real tracking situation, additions such as the time to delete a model that hasn't appeared in the scene would have to be taken into account to avoid growing the number of models in memory and increasing the processing time from comparisons.

In the scenario of multi-camera, the implementation of transition correspondence models that learn the camera configuration and predict the person pose when they enter the scene in another addition to the system that would help the re-identification algorithm.

In some cases, none of the tested features achieved good re-identification results. While there the effort was made to test as many features as possible, there are other interesting methods that could be tested such as the use of covariance matrices and different ways to store data as further dividing the region of interest in additional patches.

When dealing with multiple cameras, illumination changes are common. The Brightness Transfer Function has shown promising results in several reviewed algorithms and their addition could improve the re-identification results.

References

- [1] Paul Lewis. You're being watched: there's one CCTV camera for every 32 people in UK. <http://www.theguardian.com/uk/2011/mar/02/cctv-cameras-watching-surveillance>, March 2011. The Guardian.
- [2] James Vlahos. Surveillance society: New high-tech cameras are watching you. *Popular Mechanics*, pages 64–69, 2008.
- [3] Venkatesh Saligrama, Janusz Konrad, and Pierre-Marc Jodoin. Video anomaly identification. *Signal Processing Magazine, IEEE*, 27(5):18–33, 2010.
- [4] Cecilia Ovesdotter Alm, Dan Roth, and Richard Sproat. Emotions from text: machine learning for text-based emotion prediction. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 579–586. Association for Computational Linguistics, 2005.
- [5] Hamid Aghajan and Andrea Cavallaro. *Multi-camera networks: principles and applications*. Academic press, 2009.
- [6] Constantine P Papageorgiou, Michael Oren, and Tomaso Poggio. A general framework for object detection. In *Computer vision, 1998. sixth international conference on*, pages 555–562. IEEE, 1998.
- [7] D-N Truong Cong, Louahdi Khoudour, Catherine Achard, Cyril Meurie, and Olivier Lezou-ray. People re-identification by spectral classification of silhouettes. *Signal Processing*, 90(8):2362–2374, 2010.
- [8] Ke Chen, Shaogang Gong, and Tao Xiang. Human pose estimation using structural support vector machines. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 846–851. IEEE, 2011.
- [9] Naomi S Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992.
- [10] Marius Muja and David G Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. *VISAPP (I)*, 2, 2009.
- [11] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.
- [12] Jason Weston and Chris Watkins. Multi-class support vector machines. Technical report, Citeseer, 1998.

- [13] Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multiclass support vector machines. *Neural Networks, IEEE Transactions on*, 13(2):415–425, 2002.
- [14] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Computational learning theory*, pages 23–37. Springer, 1995.
- [15] Robert E Schapire and Yoram Singer. Boostexter: A boosting-based system for text categorization. *Machine learning*, 39(2-3):135–168, 2000.
- [16] Ron Appel, Thomas Fuchs, Piotr Dollár, and Pietro Perona. Quickly boosting decision trees-pruning underachieving features early. In *JMLR Workshop and Conference Proceedings*, volume 28, pages 594–602. JMLR, 2013.
- [17] Ludmila I Kuncheva. *Fuzzy classifier design*, volume 49. Springer Science & Business Media, 2000.
- [18] James M Keller, Michael R Gray, and James A Givens. A fuzzy k-nearest neighbor algorithm. *Systems, Man and Cybernetics, IEEE Transactions on*, (4):580–585, 1985.
- [19] Chun-Fu Lin and Sheng-De Wang. Fuzzy support vector machines. *Neural Networks, IEEE Transactions on*, 13(2):464–471, 2002.
- [20] Radu Timofte, Karel Zimmermann, and Luc Van Gool. Multi-view traffic sign detection, recognition, and 3d localisation. *Machine vision and applications*, 25(3):633–647, 2014.
- [21] Omar Javed, Khurram Shafique, and Mubarak Shah. Appearance modeling for tracking in multiple non-overlapping cameras. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 26–33. IEEE, 2005.
- [22] Fatih Porikli. Inter-camera color calibration by correlation model function. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, volume 2, pages II–133. IEEE, 2003.
- [23] Omar Javed, Khurram Shafique, Zeeshan Rasheed, and Mubarak Shah. Modeling inter-camera space-time and appearance relationships for tracking across non-overlapping views. *Computer Vision and Image Understanding*, 109(2):146–162, 2008.
- [24] Ian Jolliffe. *Principal component analysis*. Wiley Online Library, 2002.
- [25] Bryan Prosser, Shaogang Gong, and Tao Xiang. Multi-camera matching using bi-directional cumulative brightness transfer functions. In *BMVC*, volume 8, pages 164–1. Citeseer, 2008.
- [26] Unsang Park, Anil K Jain, Itaru Kitahara, Kiyoshi Kogure, and Norihiro Hagita. Vise: Visual search engine using multiple networked cameras. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 3, pages 1204–1207. IEEE, 2006.
- [27] Dong Seon Cheng, Marco Cristani, Michele Stoppa, Loris Bazzani, and Vittorio Murino. Custom pictorial structures for re-identification. In *BMVC*, volume 2, page 6. Citeseer, 2011.
- [28] Sateesh Pedagadi, James Orwell, Sergio Velastin, and Boghos Boghossian. Local fisher discriminant analysis for pedestrian re-identification. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3318–3325. IEEE, 2013.

- [29] Masashi Sugiyama. Local fisher discriminant analysis for supervised dimensionality reduction. In *Proceedings of the 23rd international conference on Machine learning*, pages 905–912. ACM, 2006.
- [30] Vildana Sulić Kenk, Stanislav Kovačič, Matej Kristan, Melita Hajdinjak, Janez Perš, et al. Visual re-identification across large, distributed camera networks. *Image and Vision Computing*, 34:11–26, 2015.
- [31] Omar Hamdoun, Fabien Moutarde, Bogdan Stanciulescu, and Bruno Steux. Person re-identification in multi-camera system by signature based on interest point descriptors collected on short video sequences. In *Distributed Smart Cameras, 2008. ICDSC 2008. Second ACM/IEEE International Conference on*, pages 1–6. IEEE, 2008.
- [32] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *Computer Vision—ECCV 2006*, pages 404–417. Springer, 2006.
- [33] Martin Bauml and Rainer Stiefelhagen. Evaluation of local features for person re-identification in image sequences. In *Advanced Video and Signal-Based Surveillance (AVSS), 2011 8th IEEE International Conference on*, pages 291–296. IEEE, 2011.
- [34] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(10):1615–1630, 2005.
- [35] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.
- [36] Chikahito Nakajima, Massimiliano Pontil, Bernd Heisele, and Tomaso Poggio. Full-body person recognition system. *Pattern recognition*, 36(9):1997–2006, 2003.
- [37] Deva Ramanan, David A Forsyth, and Andrew Zisserman. Tracking people by learning their appearance. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(1):65–81, 2007.
- [38] Yinghao Cai, Kaiqi Huang, and Tieniu Tan. Human appearance matching across multiple non-overlapping cameras. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE, 2008.
- [39] John Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6):679–698, 1986.
- [40] William Robson Schwartz and Larry S Davis. Learning discriminative appearance-based models using partial least squares. In *Computer Graphics and Image Processing (SIB-GRAPI), 2009 XXII Brazilian Symposium on*, pages 322–329. IEEE, 2009.
- [41] Robert M Haralick, Karthikeyan Shanmugam, and Its’ Hak Dinstein. Textural features for image classification. *Systems, Man and Cybernetics, IEEE Transactions on*, (6):610–621, 1973.
- [42] Michela Farenzena, Loris Bazzani, Alessandro Perina, Vittorio Murino, and Marco Cristani. Person re-identification by symmetry-driven accumulation of local features. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2360–2367. IEEE, 2010.

- [43] P-E Forssén. Maximally stable colour regions for recognition and matching. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.
- [44] Wei-Shi Zheng, Shaogang Gong, and Tao Xiang. Person re-identification by probabilistic relative distance comparison. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 649–656. IEEE, 2011.
- [45] Loris Bazzani, Marco Cristani, Alessandro Perina, and Vittorio Murino. Multiple-shot person re-identification by chromatic and epitomic analyses. *Pattern Recognition Letters*, 33(7):898–903, 2012.
- [46] Martin Hirzer, Peter M Roth, and Horst Bischof. Person re-identification by efficient impostor-based metric learning. In *Advanced Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference on*, pages 203–208. IEEE, 2012.
- [47] Li Wang and Dong-Chen He. Texture classification using texture spectrum. *Pattern Recognition*, 23(8):905–910, 1990.
- [48] Kilian Q Weinberger, John Blitzer, and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. In *Advances in neural information processing systems*, pages 1473–1480, 2005.
- [49] Alexis Mignon and Frédéric Jurie. Pcca: A new approach for distance learning from sparse pairwise constraints. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2666–2672. IEEE, 2012.
- [50] Yang Li, Ziyang Wu, and Richard J Radke. Multi-shot re-identification with random-projection-based random forests. In *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on*, pages 373–380. IEEE, 2015.
- [51] Davide Baltieri, Roberto Vezzani, and Rita Cucchiara. 3D body model construction and matching for real time people re-identification. In *Eurographics Italian Chapter Conference*, pages 65–71, 2010.
- [52] Rita Cucchiara, Costantino Grana, Andrea Prati, and Roberto Vezzani. Probabilistic posture classification for human-behavior analysis. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 35(1):42–54, 2005.
- [53] Kinh Tieu, Gerald Dalley, and W Eric L Grimson. Inference of non-overlapping camera network topology by measuring statistical dependence. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1842–1849. IEEE, 2005.
- [54] Dimitrios Makris and Tim Ellis. Automatic learning of an activity-based semantic scene model. In *2013 10th IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 183–183. IEEE Computer Society, 2003.
- [55] TJ Ellis, Dimitrios Makris, and James Black. Learning a multi-camera topology. In *Joint IEEE Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS)*, pages 165–171, 2003.
- [56] Xiaotao Zou, Bir Bhanu, Bi Song, and Amit K Roy-Chowdhury. Determining topology in a distributed camera network. In *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, volume 5, pages V–133. IEEE, 2007.

- [57] Thierry Bouwmans. Traditional and recent approaches in background modeling for foreground detection: An overview. *Computer Science Review*, 11:31–66, 2014.
- [58] Jaswant R. Jain and Anil K. Jain. Displacement measurement and its application in inter-frame image coding. *IEEE Transactions on Communications*, 29(12):1799–1808, December 1981.
- [59] Karan Gupta and Anjali V. Kulkarni. Implementation of an automated single camera object tracking system using frame differencing and dynamic template matching. *Advances in Computer and Information Sciences*, 2008.
- [60] Karan Gupta and Anjali V Kulkarni. Implementation of an automated single camera object tracking system using frame differencing and dynamic template matching. In *Advances in Computer and Information Sciences and Engineering*, pages 245–250. Springer, 2008.
- [61] R. Romano W. E. L. Grimson, C. Stauffer and L. Lee. Using adaptative tracking to classify and monitor activities in a site. *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*, pages 22–29, 1998.
- [62] Tim Ellis and Ming Xu. Object detection and tracking in an open and dynamic world. *2nd IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, 2001.
- [63] Thierry Bouwmans, Fida El Baf, and Bertrand Vachon. Background modeling using mixture of gaussians for foreground detection-a survey. *Recent Patents on Computer Science*, 1(3):219–237, 2008.
- [64] Rita Cucchiara, Costantino Grana, Massimo Piccardi, Andrea Prati, and Stefano Sirotti. Improving shadow suppression in moving object detection with HSV color information. *Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE*, pages 334–339, 2001.
- [65] Pankaj Kumar, Kuntal Sengupta, and Adrian Lee. A comparative study of different color spaces for foreground and shadow detection for traffic monitoring system. *Intelligent Transportation Systems, 2002. Proceedings. The IEEE 5th International Conference on*, pages 100–105, 2002.
- [66] Csaba Benedek and Tamás Szirányi. Bayesian foreground and shadow detection in uncertain frame rate surveillance videos. *Image Processing, IEEE Transactions on*, 17(4):608–621, 2008.
- [67] Andres Sanin, Conrad Sanderson, and Brian C Lovell. Shadow detection: A survey and comparative evaluation of recent methods. *Pattern recognition*, 45(4):1684–1695, 2012.
- [68] Olivier Barnich and Marc Van Droogenbroeck. ViBe: a powerful random technique to estimate the background in video sequences. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 945–948. IEEE, 2009.
- [69] Olivier Barnich and Marc Van Droogenbroeck. ViBe: A universal background subtraction algorithm for video sequences. *Image Processing, IEEE Transactions on*, 20(6):1709–1724, 2011.
- [70] Yin-Shi Qin, Shui-Fa Sun, Xian-Bing Ma, Song Hu, and Bang-Jun Lei. A shadow removal algorithm for ViBe in HSV color space. In *3rd International Conference on Multimedia Technology (ICMT-13)*. Atlantis Press, 2013.

- [71] Marc Van Droogenbroeck and Olivier Paquot. Background subtraction: Experiments and improvements for ViBe. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 32–37. IEEE, 2012.
- [72] Kyunghnam Kim, Thanarat H Chalidabhongse, David Harwood, and Larry Davis. Real-time foreground–background segmentation using codebook model. *Real-time imaging*, 11(3):172–185, 2005.
- [73] Michael Oren, Constantine Papageorgiou, Pawan Sinha, Edgar Osuna, and Tomaso Poggio. Pedestrian detection using wavelet templates. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 193–199. IEEE, 1997.
- [74] Paulo Menezes, José Carlos Barreto, and Jorge Dias. Face tracking based on haar-like features and eigenfaces. In *IFAC/EURON Symposium on Intelligent Autonomous Vehicles*. Citeseer, 2004.
- [75] Kobi Levi and Yair Weiss. Learning object detection from a small number of examples: the importance of good features. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–53. IEEE, 2004.
- [76] David Gerónimo, Antonio López, Daniel Ponsa, and Angel D Sappa. Haar wavelets and edge orientation histograms for on–board pedestrian detection. In *Pattern Recognition and Image Analysis*, pages 418–425. Springer, 2007.
- [77] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 1:886–893, 2005.
- [78] Hui-Xing Jia and Yu-Jin Zhang. Fast human detection by boosting histograms of oriented gradients. In *Image and Graphics, 2007. ICIG 2007. Fourth International Conference on*, pages 683–688. IEEE, 2007.
- [79] Timo Ahonen, Abdenour Hadid, and Matti Pietikainen. Face description with local binary patterns: Application to face recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(12):2037–2041, 2006.
- [80] Xiaoyu Wang, Tony X Han, and Shuicheng Yan. An HOG-LBP human detector with partial occlusion handling. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 32–39. IEEE, 2009.
- [81] Shanshan Zhang, Christian Bauckhage, and Armin B Cremers. Informed Haar-like features improve pedestrian detection. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 947–954. IEEE, 2014.
- [82] Wanli Ouyang and Xiaogang Wang. A discriminative deep model for pedestrian detection with occlusion handling. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3258–3265. IEEE, 2012.
- [83] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. Monocular 3D pose estimation and tracking by detection. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 623–630. IEEE, 2010.

- [84] Keinosuke Fukunaga and Larry Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *Information Theory, IEEE Transactions on*, 21(1):32–40, 1975.
- [85] Yizong Cheng. Mean shift, mode seeking, and clustering. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(8):790–799, 1995.
- [86] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Real-time tracking of non-rigid objects using mean shift. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 142–149. IEEE, 2000.
- [87] Gary R Bradski. Computer vision face tracking for use in a perceptual user interface. 1998.
- [88] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering*, 82(1):35–45, 1960.
- [89] Simon J Julier and Jeffrey K Uhlmann. A new extension of the Kalman filter to nonlinear systems. In *Int. symp. aerospace/defense sensing, simul. and controls*, volume 3, pages 3–2. Orlando, FL, 1997.
- [90] Neil J Gordon, David J Salmond, and Adrian FM Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. In *IEE Proceedings F (Radar and Signal Processing)*, volume 140, pages 107–113. IET, 1993.
- [91] Kenji Okuma, Ali Taleghani, Nando De Freitas, James J Little, and David G Lowe. A boosted particle filter: Multitarget detection and tracking. In *Computer Vision-ECCV 2004*, pages 28–39. Springer, 2004.
- [92] Jaco Vermaak, Arnaud Doucet, and Patrick Pérez. Maintaining multimodality through mixture tracking. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1110–1116. IEEE, 2003.
- [93] Raquel Urtasun, David J Fleet, and Pascal Fua. 3D people tracking with gaussian process dynamical models. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 238–245. IEEE, 2006.
- [94] Raúl Mohedano, Carlos R del Blanco, Fernando Jaureguizar, Luis Salgado, and Narciso García. Robust 3D people tracking and positioning system in a semi-overlapped multi-camera environment. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 2656–2659. IEEE, 2008.
- [95] Aniket Bera, Nico Galoppo, Dillon Sharlet, Adam Lake, and Dinesh Manocha. Adapt: real-time adaptive pedestrian tracking for crowded scenes. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 1801–1808. IEEE, 2014.
- [96] Ben Benfold and Ian Reid. Stable multi-target tracking in real-time surveillance video. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3457–3464. IEEE, 2011.
- [97] Boris Babenko, Ming-Hsuan Yang, and Serge Belongie. Visual tracking with online multiple instance learning. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 983–990. IEEE, 2009.

- [98] Thomas G Dietterich, Richard H Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence*, 89(1):31–71, 1997.
- [99] CAVIAR test case scenarios. <http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/>, January 2004.
- [100] Adam Nilski. Evaluating multiple camera tracking systems-the i-lids 5 th scenario. In *2008 42nd Annual IEEE International Carnahan Conference on Security Technology*, 2008.
- [101] Andreas Ess, Bastian Leibe, and Luc Van Gool. Depth and appearance for mobile scene analysis. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.
- [102] PETS 2001 dataset. <http://www.cvg.reading.ac.uk/PETS2001/pets2001-dataset.html>, 2001.
- [103] Taiqing Wang, Shaogang Gong, Xiatian Zhu, and Shengjin Wang. Person re-identification by video ranking. In *Computer Vision–ECCV 2014*, pages 688–703. Springer, 2014.
- [104] D. Gray, S. Brennan, and H. Tao. Evaluating Appearance Models for Recognition, Reacquisition, and Tracking. *Proc. IEEE International Workshop on Performance Evaluation for Tracking and Surveillance (PETS)*, 2007.
- [105] Roberto Vezzani and Rita Cucchiara. Video surveillance online repository (visor): an integrated framework. *Multimedia Tools and Applications*, 50(2):359–380, 2010.
- [106] Loris Bazzani. *Beyond Multi-target Tracking*. PhD thesis, PhD thesis, Verona, ITALY, 2012.
- [107] David Martin Powers. Evaluation: from precision, recall and f-measure to ROC, informedness, markedness and correlation. 2011.
- [108] Roberto Vezzani, Davide Baltieri, and Rita Cucchiara. People reidentification in surveillance and forensics: A survey. *ACM Computing Surveys (CSUR)*, 46(2):29, 2013.
- [109] Dimitri A Lisin, Marwan A Mattar, Matthew B Blaschko, Erik G Learned-Miller, and Mark C Benfield. Combining local and global image features for object class recognition. In *Computer Vision and Pattern Recognition-Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference on*, pages 47–47. IEEE, 2005.
- [110] Shamik Sural, Gang Qian, and Sakti Pramanik. Segmentation and histogram generation using the hsv color space for image retrieval. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 2, pages II–589. IEEE, 2002.
- [111] Riccardo Mazzon, Syed Fahad Tahir, and Andrea Cavallaro. Person re-identification in crowd. *Pattern Recognition Letters*, 33(14):1828–1837, 2012.
- [112] K-E Aziz, Djamel Merad, and Bernard Fertil. People re-identification across multiple non-overlapping cameras system by appearance classification and silhouette part segmentation. In *Advanced Video and Signal-Based Surveillance (AVSS), 2011 8th IEEE International Conference on*, pages 303–308. IEEE, 2011.
- [113] Jianxin Wu and Jim M Rehg. Centrist: A visual descriptor for scene categorization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(8):1489–1501, 2011.

- [114] Ramin Zabih and John Woodfill. Non-parametric local transforms for computing visual correspondence. In *Computer Vision—ECCV’94*, pages 151–158. Springer, 1994.
- [115] Michael Oren, Constantine Papageorgiou, Pawan Sinha, Edgar Osuna, and Tomaso Poggio. Pedestrian detection using wavelet templates. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 193–199. IEEE, 1997.
- [116] Icaro Oliveira de Oliveira and José Luiz de Souza Pio. People reidentification in a camera network. In *Dependable, Autonomic and Secure Computing, 2009. DASC’09. Eighth IEEE International Conference on*, pages 461–466. IEEE, 2009.
- [117] Slawomir Bak, Etienne Corvee, Francois Brémont, and Monique Thonnat. Person reidentification using haar-based and dcd-based signature. In *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*, pages 1–8. IEEE, 2010.
- [118] Michel Lantagne, Marc Parizeau, and Robert Bergevin. Vip: Vision tool for comparing images of people. In *Vision Interface*, volume 2, 2003.
- [119] Raman Maini and Himanshu Aggarwal. Study and comparison of various image edge detection techniques. *International journal of image processing (IJIP)*, 3(1):1–11, 2009.
- [120] David J Heeger and James R Bergen. Pyramid-based texture analysis/synthesis. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 229–238. ACM, 1995.
- [121] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *Computer Vision—ECCV 2006*, pages 430–443. Springer, 2006.
- [122] Pedro Carvalho, Telmo Oliveira, Lucian Ciobanu, Filipe Gaspar, Luís F Teixeira, Rafael Bastos, Jaime S Cardoso, Miguel S Dias, and Luís Côrte-Real. Analysis of object description methods in a video object tracking environment. *Machine vision and applications*, 24(6):1149–1165, 2013.
- [123] Steve Baker and Robert D Cousins. Clarification of the use of chi-square and likelihood functions in fits to histograms. *Nuclear Instruments and Methods in Physics Research*, 221(2):437–442, 1984.
- [124] Sung-Hyuk Cha. Comprehensive survey on distance/similarity measures between probability density functions. *City*, 1(2):1, 2007.
- [125] Bryan Prosser, Wei-Shi Zheng, Shaogang Gong, Tao Xiang, and Q Mary. Person reidentification by support vector ranking. In *BMVC*, volume 2, page 6, 2010.

